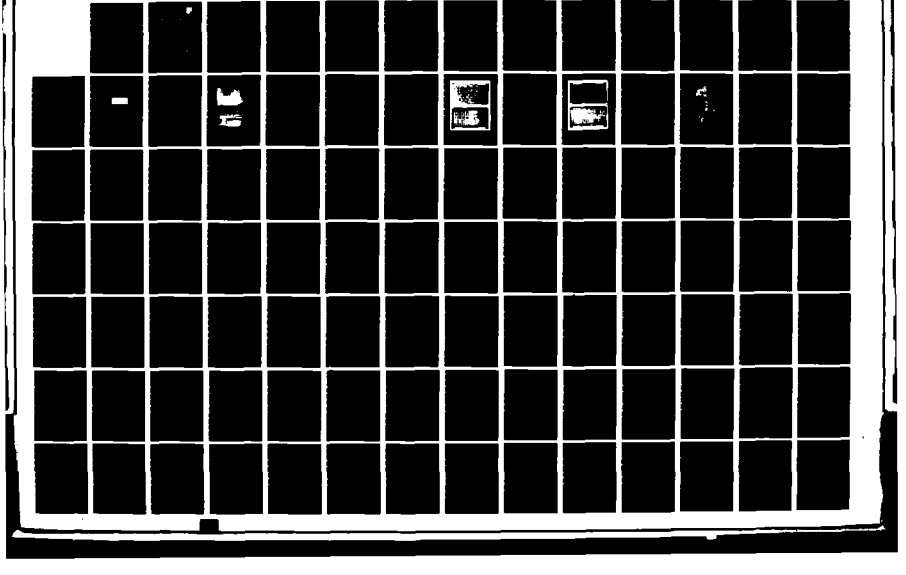
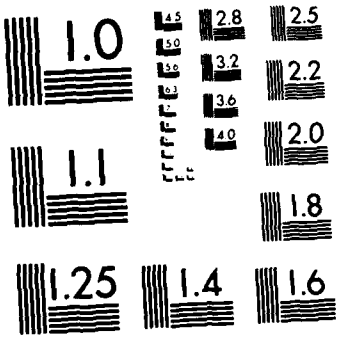
AD-A134 730 S-BAND RANGE TRACKER AND SURVEILLANCE LAB INTERFACE(U) 1/2
ROME AIR DEVELOPMENT CENTER GRIFFISS AFB NY B D BUSH
SEP 83 RADC-TR-83-213

UNCLASSIFIED F/G 17/9 NL





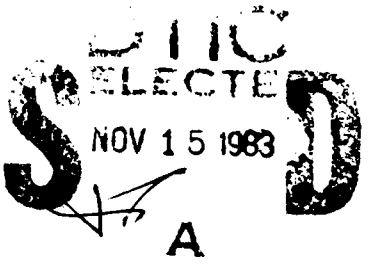
RADC-TR-83-213
In-House Report
September 1983

(12)

S-BAND RANGE TRACKER AND SURVEILLANCE LAB INTERFACE

Brian D. Bush

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED



ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-83-213 has been reviewed and is approved for publication.

APPROVED:

FRED J. DEMMA
Chief, Surveillance Technology Branch
Surveillance Division

APPROVED:

FRANK J. REHM
Chief, Surveillance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (OCTS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-83-213	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) S-BAND RANGE TRACKER AND SURVEILLANCE LAB INTERFACE		5. TYPE OF REPORT & PERIOD COVERED In-House Report 6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Brian D. Bush		8. CONTRACT OR GRANT NUMBER(s) N/A
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rome Air Development Center (OCTS) Griffiss AFB NY 13441		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45061158
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (OCTS) Griffiss AFB NY 13441		12. REPORT DATE September 1983 13. NUMBER OF PAGES 112
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

S-Band tracking radar
Algorithms
System interfaces

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents the design, construction, test and laboratory integration of the range tracker and associated subsystems for the RADC/OC Surveillance Laboratory's S-Band tracking radar. This development was accomplished over the period from December 1981 to November 1983 and was designed, constructed and tested entirely in-house. This report contains information on the use of the range tracker, its interfaces to other laboratory equipment, the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

philosophy behind its design, the detailed design of the hardware (including schematics, wiring and cabling diagrams), the detailed software design (including flowcharts), and the mathematical description of its algorithms. The range tracker will be used in conjunction with other equipment in the OC Surveillance Lab in the taking and recording of radar data during flight tests.

TABLE OF CONTENTS

A-1

LIST OF FIGURES

LIST OF TABLES

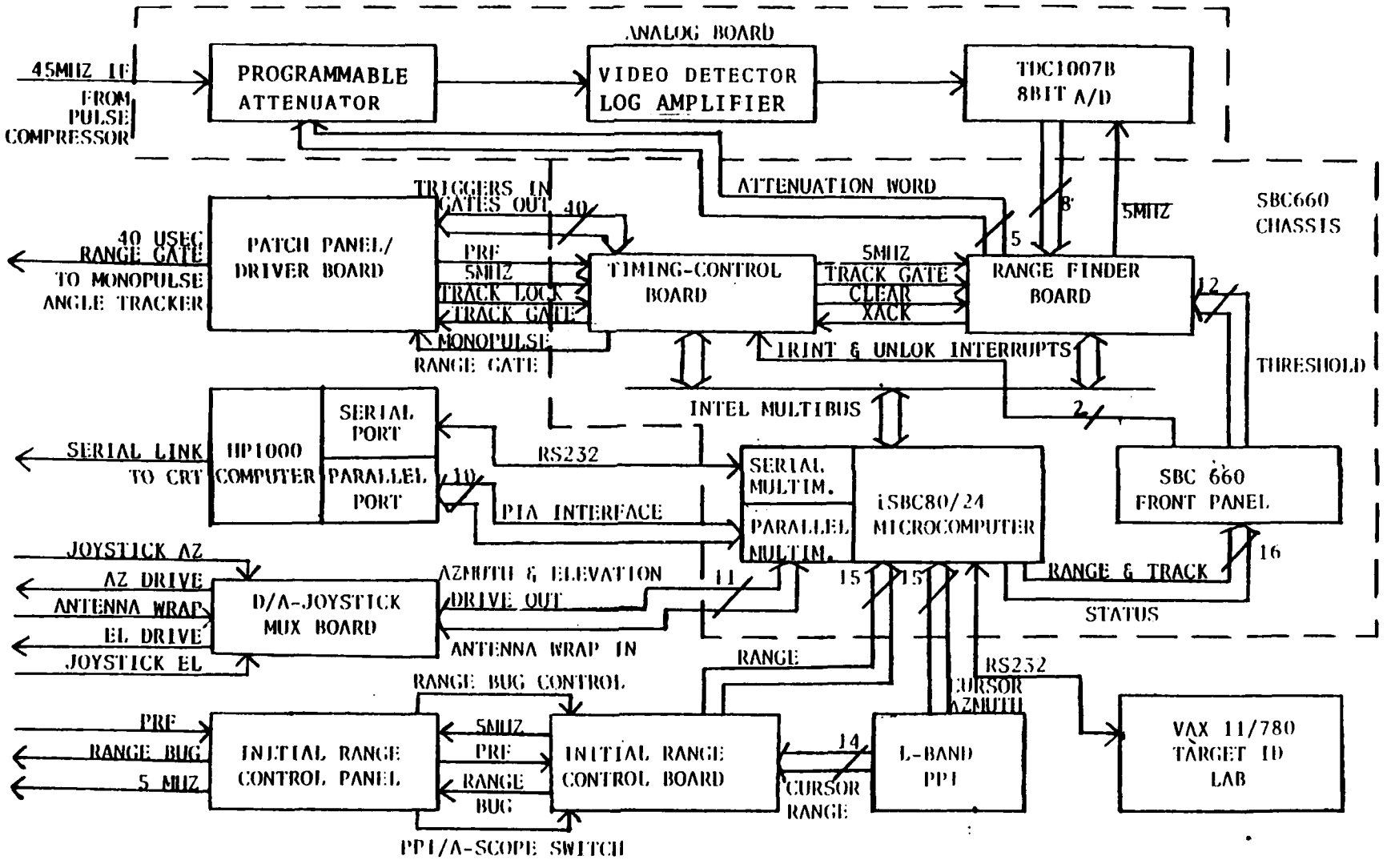
1. INTRODUCTION

This report concerns the in-house design, construction and system integration of the range tracker, programmable gates and other associated functions for the S-band tracking radar. The basic functions of this system are to provide a range gate for the monopulse angle tracker, to provide range information and tracking status to the Surveillance Laboratory's Hewlett Packard (HP) computer for recording purposes and to provide general purpose programmable gates that can be easily reconfigured. A block diagram of the range tracker and its system interfaces are shown in figure 1. This subsystem consists of an Intel iSBC 80/24 single board microcomputer with serial and parallel multimodule expansion boards, five digital boards, one analog board, an Intel SBC 660 chassis, cables and control software. The one analog board converts the IF output from the pulse compressor into 8-bit digital words. The range finder board takes this stream of 8-bit words and finds the position and value of the highest peak in the tracking (range) gate and the values of the peaks of adjacent samples. The computer then takes this information and estimates the true range of the target to 1/6 of a range cell (15 meters).

The Timing-Control board generates all gates necessary for the monopulse system and provides 20 programmable gates for system timing purposes. Timing information for the gates is downloaded to the micro over a serial link from the HP computer. The operation of these gates is controlled using an interactive program called "GATES" which is executed on the HP computer. The programmable gates and their trigger inputs are accessed at a patch panel along with the range gate outputs from the tracker. This panel also contains jacks for the 3 system inputs: PRF, angle track locked signal and the 5 MHz system clock. All of the programmable gate outputs and range gate outputs are buffered on the driver board with 50 ohm line drivers before going to the jacks. The driver board is mounted on the backside of the patch panel.

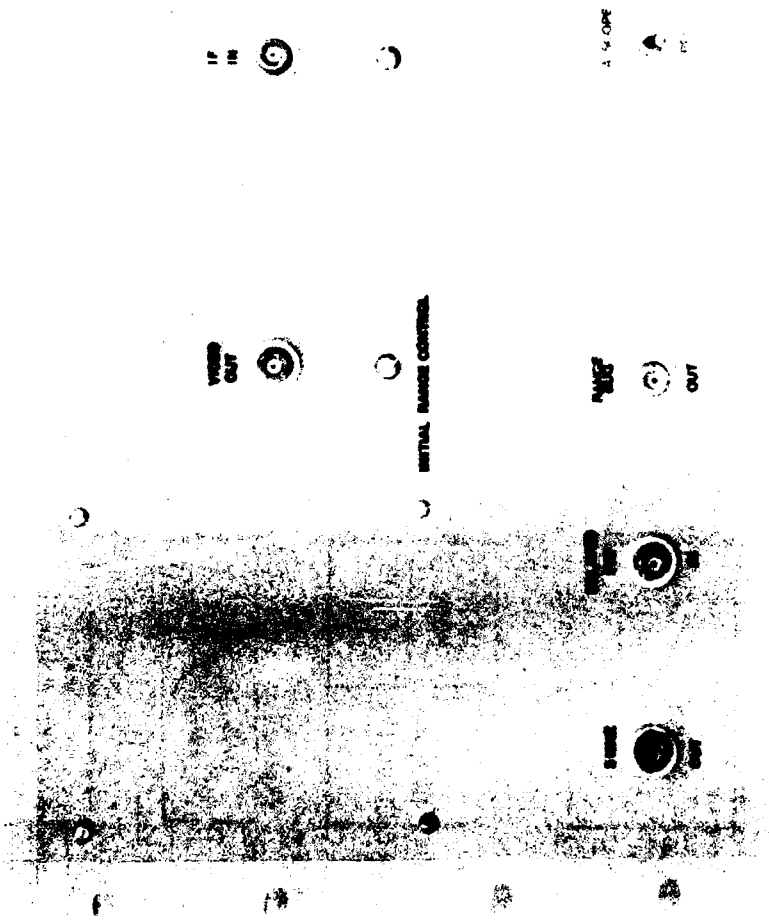
The Initial Range Control board provides the tracker with the initial range information that will be used to initiate a track. This board is controlled via the initial range control panel that allows the operator to select between two different initial range sources; the L-band PPI or A-scope. If the initial range is taken from the PPI the micro will also read the azimuth position of the PPI cursor and will drive the S-band antenna to this azimuth. The micro drives the antenna using the D/A-Joystick MUX board which is mounted in the S-band antenna control rack.

The microcomputer: provides system control; implements an Alpha-Beta tracking filter; provides gain control via a programmable attenuator; displays range and tracking status on the front panel of the 660 chassis; reports range and tracking status to the HP computer; accepts S and L-band antenna position from the HP; interfaces with the HP to download information to initialize the programmable gates; accepts initial range information from the L-band PPI or A-scope; accepts azimuth position from the L-band PPI or HP computer and then drives the S-band antenna to the azimuth received; interfaces with the VAX computer to provide azimuth and range information to target ID programs.



RANGE TRACKER / SYSTEM INTERFACE

FIGURE 1



INITIAL RANGE CONTROL PANEL
FIGURE 2

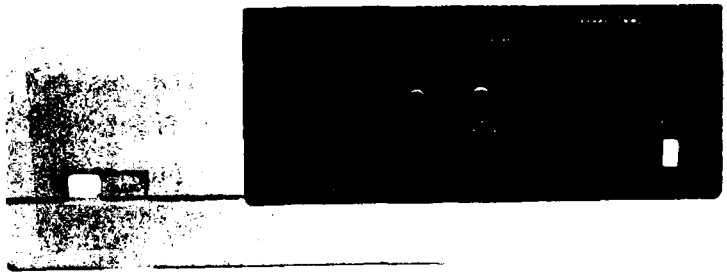
2. OPERATION

a. THE RANGE TRACKER

The range tracker was designed to accept an initial range from the L-band PPI display or an in-house built initial range control board that allows range information to be extracted from an A-scope. One of these two initial range sources can be selected at the initial range control panel shown in figure 2. If the PPI is selected, the initialization would start with both the L-band radar and S-band radar operating. The L-band radar is used initially to find the target in search mode, then the operator would place the PPI cursor over the desired target and press the initial range control button on the front of the range tracker. This causes the tracker to read the BCD range and azimuth information from the PPI. The tracker takes the initial range and displays it on the front panel (figure 3), it sets up the wide tracking gate (8.2nm) and the 40 usec monopulse gate around the indicated range and slews the S-band antenna to the proper azimuth. When the antenna stops moving, the tracker will turn on the "STANDBY" indicator light on the front panel. The range tracker then sits and waits for the angle tracker to find the target.

If the A-scope were selected for initial range information the operator would position the range bug over the desired target on the A-scope. This is done using the range bug control mounted on the initial range control panel. Once the bug is placed over the desired target the "INITIAL RANGE" button is pushed on the front panel of the range tracker, this causes the same results as described when the PPI is used for initiation except that the antenna is not moved.

When the angle tracker has found the target it indicates this to the range tracker by pulsing the angle track lock line. During the time between tracker initialization and the angle track locked signal the range tracker does not try to track the target. It assumes the target to be in a fixed position and generates a gate for the angle tracker.

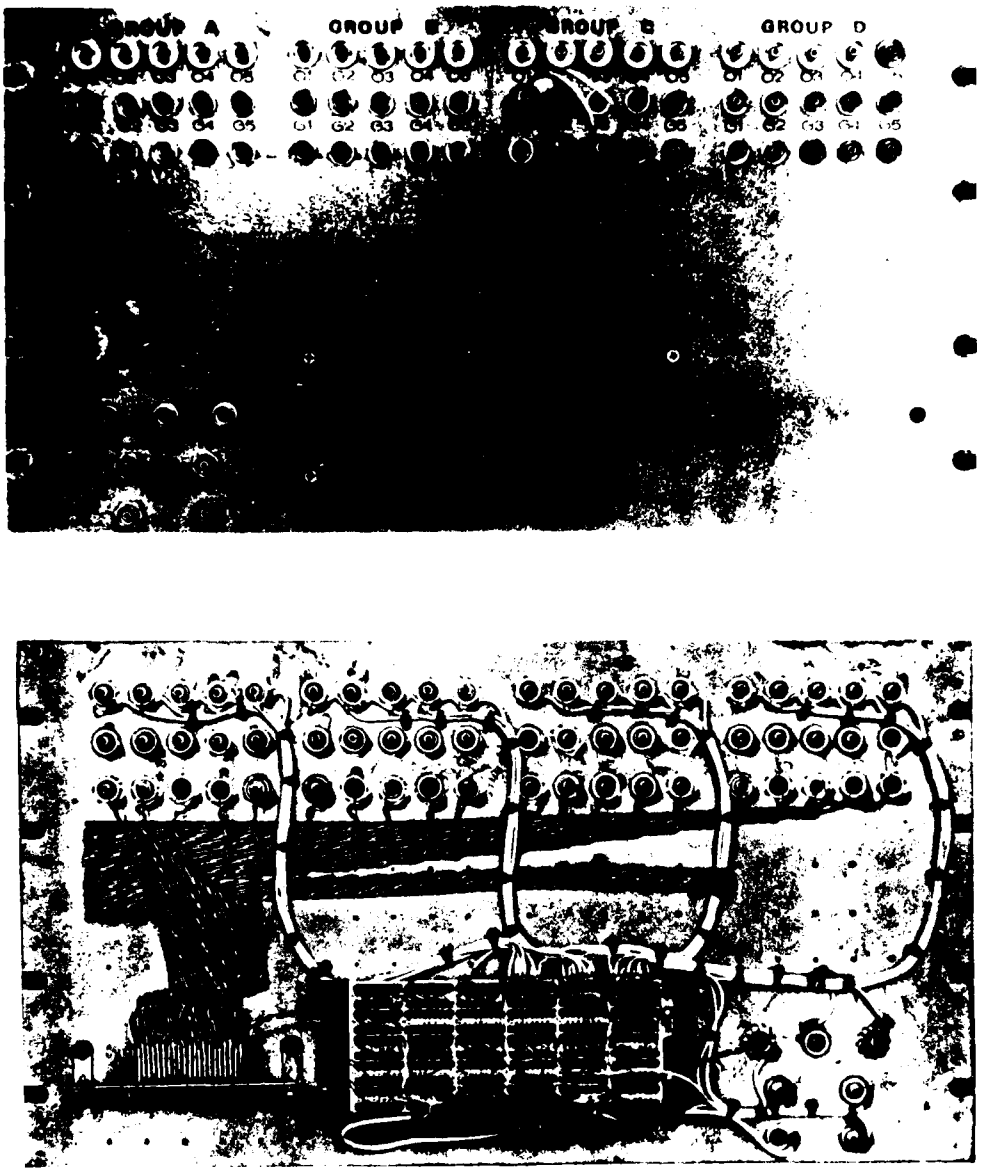


RANGE TRACKER FRONT PANEL
FIGURE 3

Once the angle track locked signal is received the range tracker will begin tracking. The Alpha-Beta tracking filter (described in detail in a latter section) requires a two second period to initialize. During the first two seconds the filter is watching the target, gathering information on its position and velocity. During this time the wide tracking gate (8.2 nm) is used. After this two second period the tracker begins to predict the average position of the target over the next one second period. Depending on the error in this prediction the wide gate, the medium size gate (3.1 nm) or the small gate (1.5 nm) will be used. The width of the gate being used is a measure of how well the tracker is tracking the target. If the small gate is being used this is indicated on the front panel by the "TRACK LOCKED" light. If the tracker is using one of the larger gates the "STANDBY" indicator will be lit.

There are two features of the tracker which should now be explained. The first is called "COASTING" and is indicated on the front panel by a lit "COASTING" indicator light. Coasting will occur if a return of greater amplitude than the target being tracked enters into the tracking gate. (ex. target moving into chaff or clutter, another target moving into the range window, etc.). If this happens the range tracker will ignore all returns, it will open up the tracking gate to the large 50 usec gate and will move it and the monopulse gate at the known velocity of the target. The tracker will continue in this mode until the "COAST UNLOCK" button is pressed on the front panel, at which time it will reacquire the target. This mode of operation is needed to allow the tracker to maintain a good track through strong chaff or clutter, or to keep it from locking onto a stronger aircraft which has entered into the range window.

The second feature of the tracker comes into play if the target being tracked falls out or drops drastically in amplitude (target scintillation, angle track lost). If this happens the algorithm will restart itself by re-setting Alpha=1 and Beta=0, the tracker will gradually increase the tracking gate size and will decrease the



PATCH PANEL / DRIVER BOARD
FIGURE 1

attenuation of the incoming signal in an attempt to find the target. The two range gates are moved at the last known velocity until the target peaks up, at which time the tracker can reacquire the target and continue with a normal tracking procedure. During this period of time the "STANDBY" indicator will be lit.

b. THE PROGRAMMABLE GATES

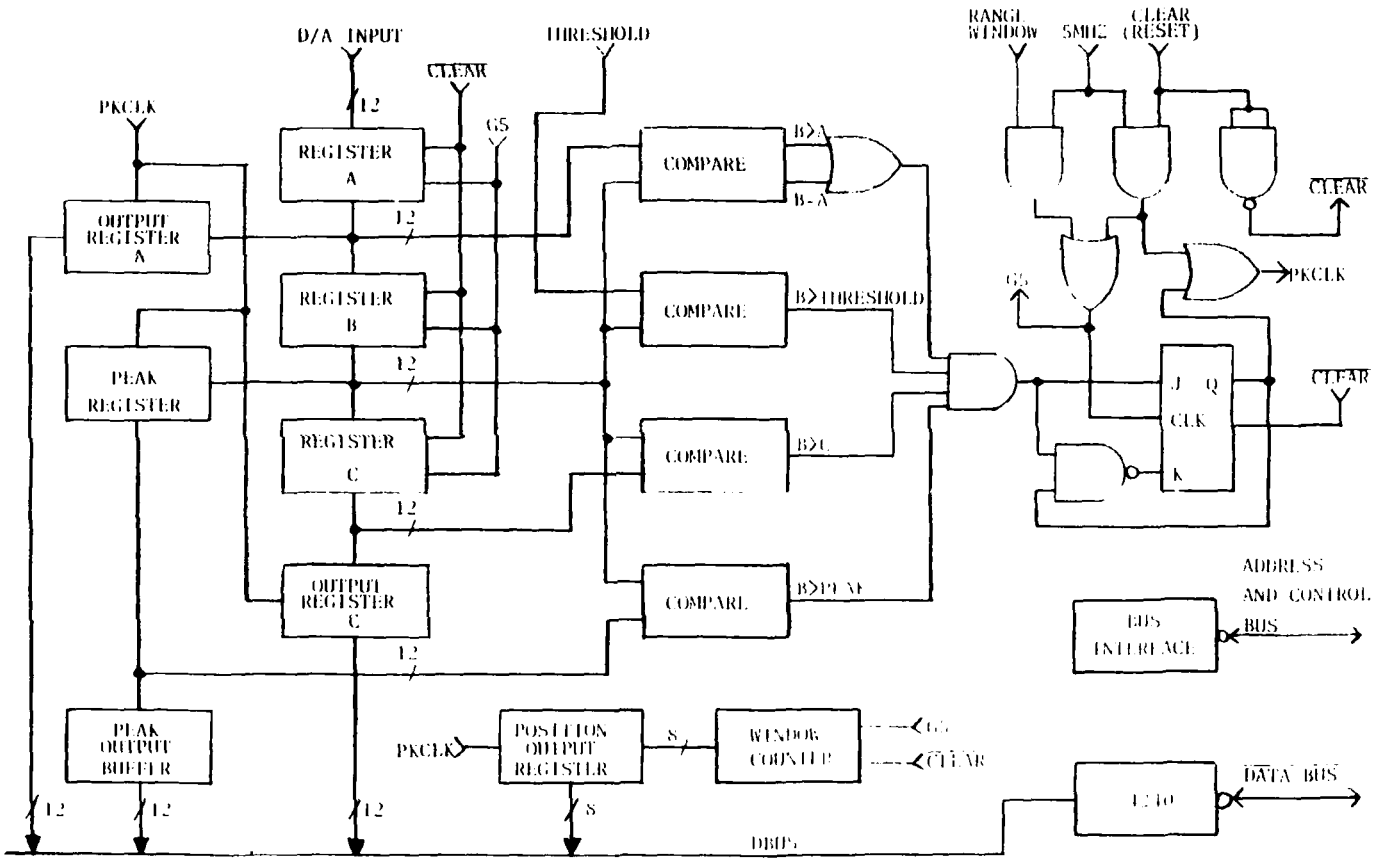
The initialization and operation of the programmable gates are accomplished via any CRT connected to the SL's HP computer. Once logged in, the operator would execute a program called "GATES". This program allows the user to initialize or modify the operation of any of the 20 programmable gates through a menu-answer type of interaction. The status of each of the gates, as specified using the "GATES" program, is saved in a file on the HP computer called "GATEST". This file contains the operating mode and timing information associated with each of the 20 gates. It also contains a 16 character label given to each gate during initialization. Gates that have not been initialized are given a label of "NOT INITIALIZED" when the gate status is listed. Downloading is also accomplished using the same program. In order for the downloader to work the microcomputer must be operating, if it is not the program will indicate this. Downloading must be done after power up and also after modifying a gate to obtain the desired result.

The patch panel, shown in figure 4, contains all the trigger input jacks for the 20 programmable gates. It also contains the jacks for the range gate outputs and system inputs. The inputs include a 5 MHz system clock, PRF, and the angle track locked signal. The 20 gates are broken down into 4 groups (A thru D), corresponding to 4 AM9513 system timing controllers used to generate the gates. Each group contains five gate outputs labeled O1 thru O5 and five corresponding trigger inputs labeled G1 thru G5. The second and third row jacks are common to make connections easier if a single trigger is required for more than one gate. The two range gate outputs on the panel move as the target moves. The gate labeled "PC" is the gate used by the range tracker to track the target. It gates out the compressed pulse of the target and is delayed

124 usec from the 40 usec range gate to allow for the compression delay. The width of the PC gate depends on the error in the tracking algorithm. The gate labeled "40 USEC" is used by the monopulse angle tracker to gate out the uncompressed 40 usec pulse. This gate is centered around the desired range and remains the same width regardless of the tracking error. These two gates are the only gates provided that move with the target. They should be used as trigger inputs to any of the programmable gates that are needed for tracking purposes. All outputs provided on the patch panel are buffered with 50 ohm line drivers including the 5 MHz and PRF outputs.

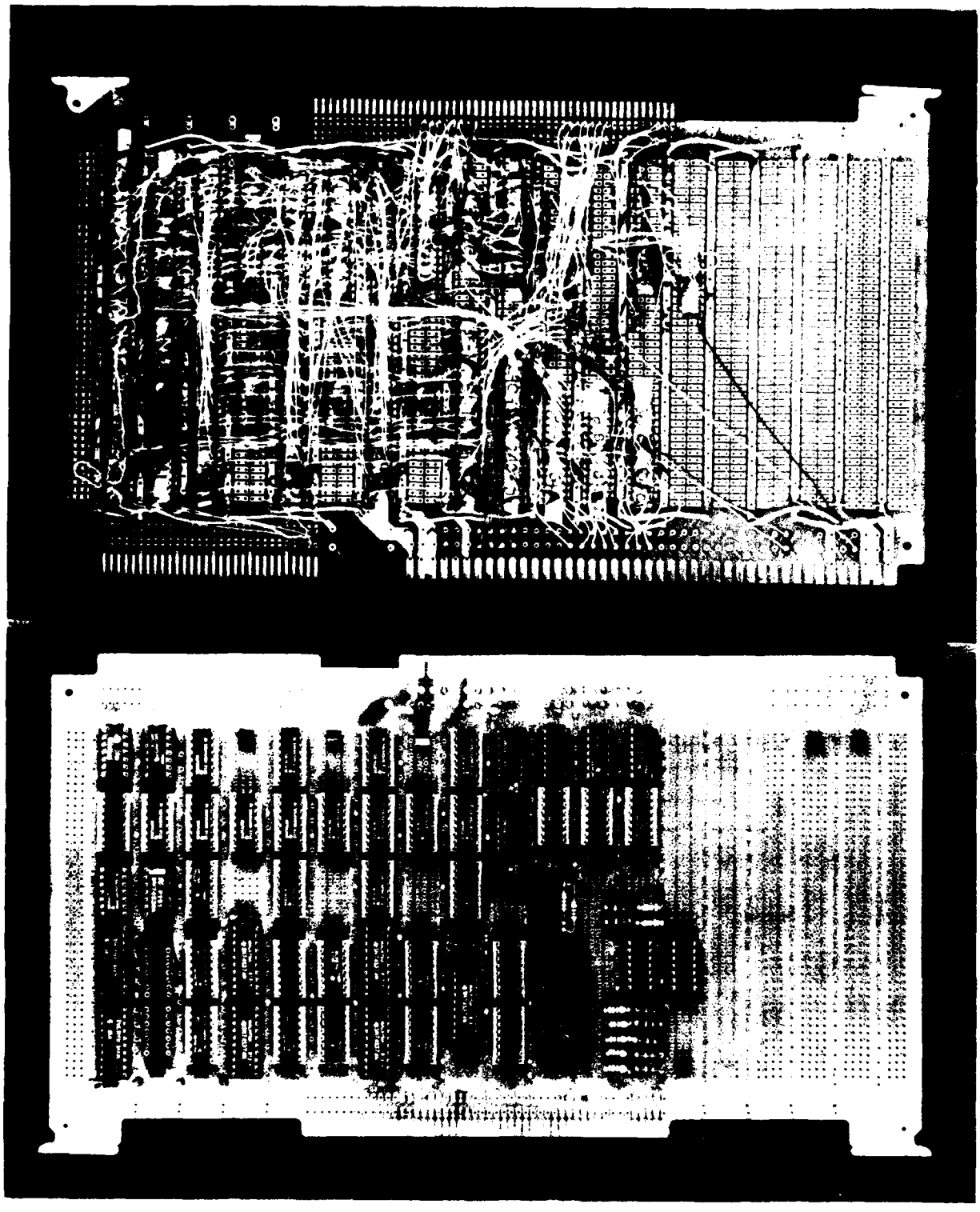
c. TARGET ID LAB/VAX INTERFACE

This interface provides the target ID lab's VAX computer with range and azimuth information from the L-band PPI cursor. This information is needed by target ID programs to establish a track on a target before any interrogation can be done. The operation of this interface starts with the L-band radar operating. The PPI operator would choose a target on the screen and place the PPI cursor over it. The "TARGET ID" button mounted on the PPI would then be pushed, thus causing the microprocessor to read the range and azimuth position of the cursor from the PPI. The micro converts this parallel BCD input into an ASCII string which it saves in a buffer for transfer to the VAX. Also contained in this buffer is a flag which is used to determine the status of the buffer, this flag is set to "1" when the buffer is filled. After this buffer is sent to the VAX this flag is reset to indicate that the data present in the buffer has aleady been transferred. This flag should be used by the VAX software to discriminate between new and old data. The VAX initiates the buffer transfer by sending a character to the micro. On receiving this character an interrupt is generated in the micro which invokes a service routine to transfer the buffer and reset the status flag. The format in which the buffer is sent is shown in table 3.



BLOCK DIAGRAM - RANGE FINDER BOARD

FIGURE 5



RANGE FINDER BOARD

FIGURE 6

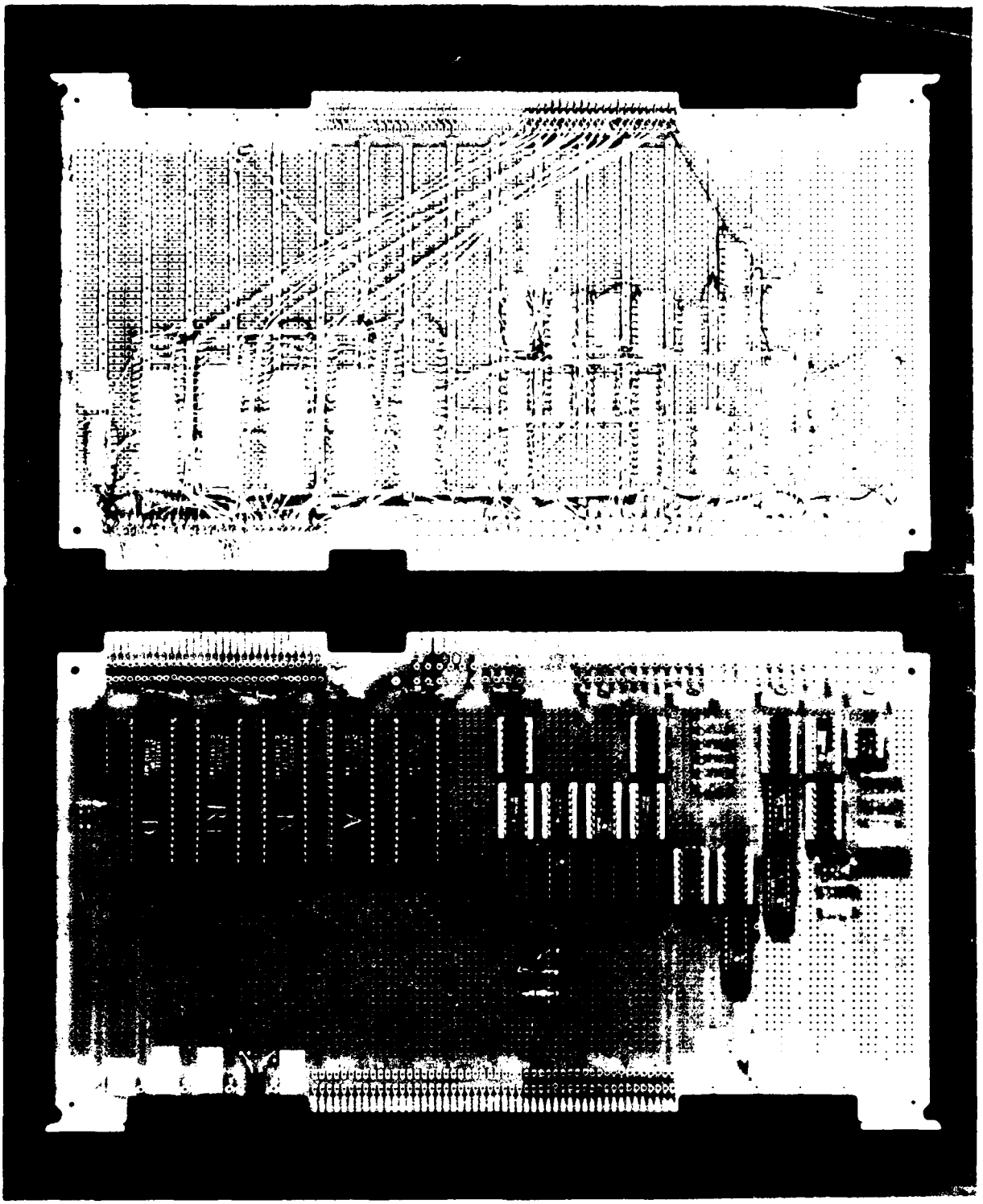
3. HARDWARE

a. THE RANGE FINDER BOARD

The range finder board (shown in figures 5 and 6) finds the value and position of the highest peak in the detection window gate and the values of its two adjacent samples. The computer uses these values in estimating the range of the target to 1/6 of a range cell (0.1 usec). While the detection window gate is high the samples from the A/D converter are clocked into registers A,B and C at a 5 MHz rate. During the time between samples the value in register B is compared to its two adjacent samples (registers A and C), to the value of the last peak found (zero initially) and to a threshold value which is controlled by the thumbwheel switch on the front panel. If the value in register B is found to be greater than register C, the threshold, and the value of the last peak found and also greater than or equal to register A then the value in register B is a peak value. If a peak is found the "PKSTB" line is enabled which allows the values of the three registers and the value of the window counter to be clocked into the output registers on the next rising edge of the 5 MHz clock. These values will remain in the output registers for the remainder of the detection window gate unless a higher peak is found. When the detection window gate goes low an interrupt is triggered in the microprocessor and these values are read and a gate is enabled which clears and resets the board for the next PRF. (Schematics and timing diagrams are shown in appendix A.)

b. THE TIMING-CONTROL BOARD

The timing-control board (figure 7) contains the hardware necessary for generating the 20 programmable gates, the 40 usec monopulse tracking gate and the detection window gate and clear pulse necessary for control of the range finder board. It also contains the hardware to generate all the various interrupts necessary (Initial Range (IRINT), End of Range Window (EWINT),One Second Update (UPDAT), Coasting Unlock (UNLOK)) for control of the tracker and provides a handshake signal (XACK) to the



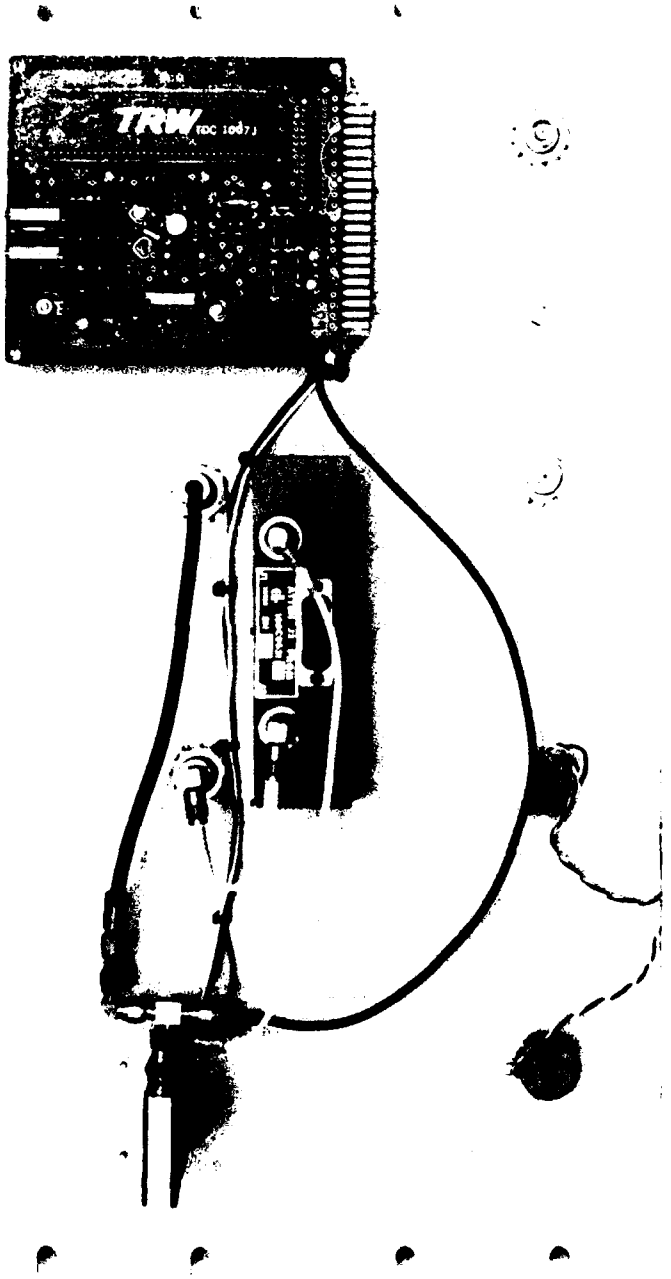
TIMING - CONTROL BOARD
FIGURE 7

micro. All timing functions are done using Advanced Micro Device's AM9513 System Timing Controllers. Each of these chips contain 5 programmable gates each with its own triggering input. There are 5 of these chips on the board, the first, labeled "RT", is dedicated to the range tracker for internal control and generation of the range gates. The rest, labeled "A" thru "D", are used to generate the 20 programmable gates and can be programmed to operate in any one of the following modes:

1. DELAYED PULSE ONESHOT, RISING EDGE TRIGGERED, ACTIVE HIGH PULSE
2. DELAYED PULSE ONESHOT, FALLING EDGE TRIGGERED, ACTIVE HIGH PULSE
3. DELAYED PULSE ONESHOT, RISING EDGE TRIGGERED, ACTIVE LOW PULSE
4. DELAYED PULSE ONESHOT, FALLING EDGE TRIGGERED, ACTIVE LOW PULSE
5. DUTY CYCLE CLOCK, TRIGGER HIGH ENABLED
6. DUTY CYCLE CLOCK, TRIGGER LOW ENABLED
7. DUTY CYCLE CLOCK, NO HARDWARE GATING

Each of the 5 programmable gates in one AM9513 consists of two 16 bit registers, a 16 bit up/down counter and control logic that allows the counter to operate in several different modes. The basic operation is to load the counter from one of the 16 bit registers and to start a down count when the trigger is applied. When the counter reaches zero the output flipflop is toggled, the counter is reloaded from the other register and the cycle starts again. Thus, by varying the amounts in either register, various duty cycles can be obtained.

The outputs of the programmable gates and their triggering inputs are all brought out to a patch panel for easy access. Each of the gate outputs and the two range window gates are buffered with 50 ohm line drivers before going to the connectors. The board which contains these drivers is attached to the back of the patch panel. (Schematics and timing diagrams for the Timing- Contol board and the Driver board are in appendix A.)



ANALOG BOARD
FIGURE 8

c. ANALOG BOARD

The one analog board (shown in figure 8) consists of a video detector/log amplifier, a 0 to 40 dB programmable attenuator and an 8 bit video A/D converter. This board converts the IF output of the pulse compressor to 8 bit digital words which can then be interpreted by the Range Finder board. This hardware is mounted on the same panel as the initial range controls. There is an input jack for the IF and an output jack for the detected video. There is also a hole in the panel where the baseline adjustment for the video detector can be accessed. The video baseline should be adjusted to zero volts before attempting to track.

d. D/A JOYSTICK MUX BOARD

This board contains two 4 bit D/A converters and some multiplexing circuitry to allow the micro to drive the S-band antenna in both azimuth and elevation, it also contains the circuitry which provides the micro with a digital representation of the wrap-up of the antenna. The multiplexer switch is controlled by the micro which switches the input to the antenna drive circuitry between the D/A's and the joystick. It is so designed that during power off on the micro the joystick has control over the antenna. Baseline adjustments and gain controls are provided on the board for each D/A converter and gain controls are provided for the joystick azimuth and elevation inputs. The maximum output drive to the antenna is limited by the analog switches (multiplexer) to +5 volts. This limits the maximum drive speed to 12 degrees per second. This board is mounted in the S-band antenna control rack. (Schematics are in appendix A.)

e. INITIAL RANGE CONTROL BOARD

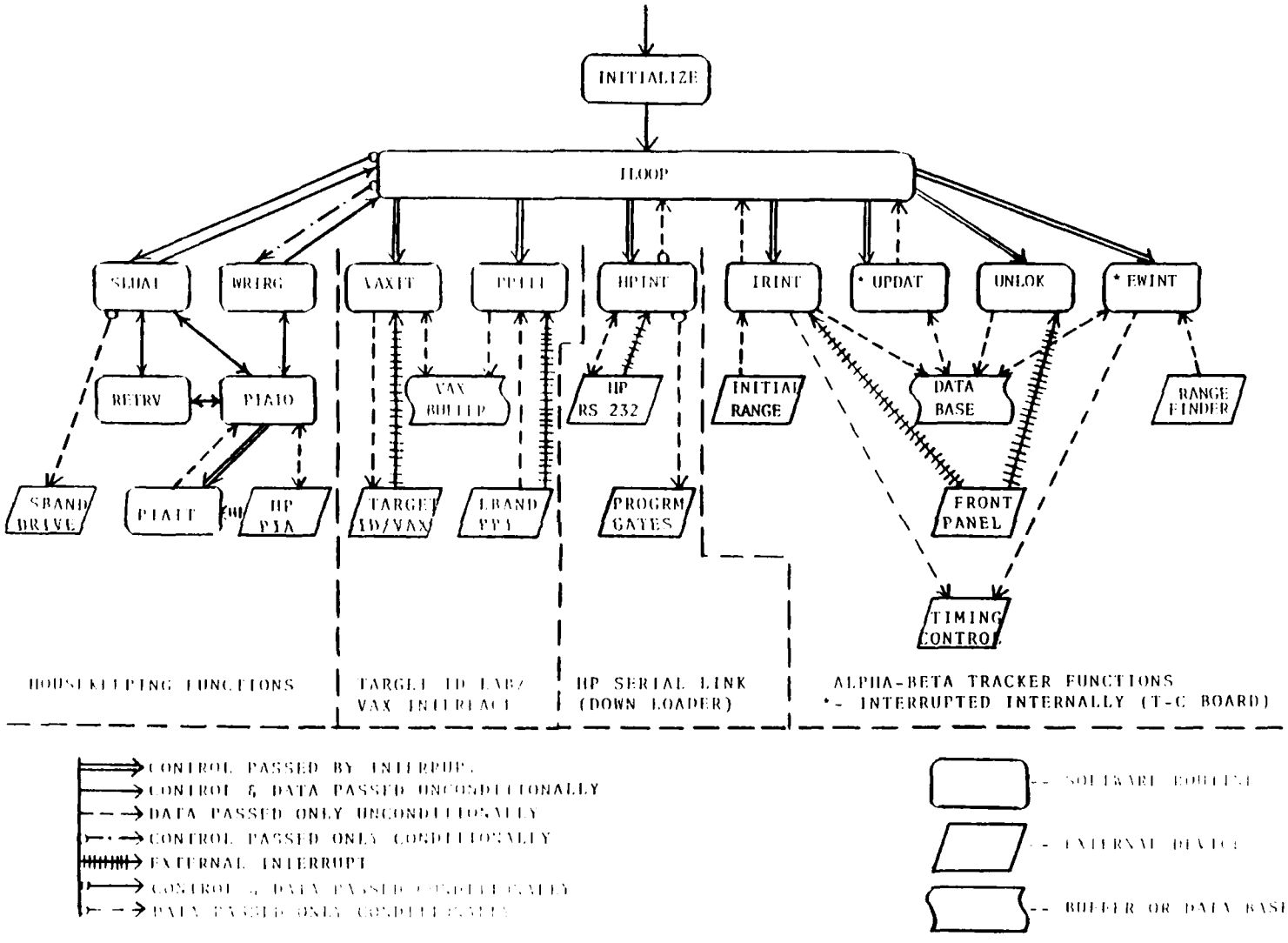
The initial range control board provides the means to choose between two initial range sources, the L-band PPI or the A-scope. This board contains the circuitry necessary to extract range information from an A-scope. It produces a 5 volt, 2 usec pulse ("RANGE BUG") which can

be moved in range using the "RANGE BUG CONTROL" on the control panel. The range of the "bug" is estimated by counting the delay from the PRF to the bug in tenth mile increments (1.2 usec). This range estimate is then routed to a multiplexer along with the BCD range input from the L-band PPI. The operator can then select between the two sources using the switch on the initial range control panel.

This board accepts one input and provides two outputs to the initial range control panel. One input is the delayed PRF that is used as the zero range mark, this is the same PRF used to trigger the A-scope and is delayed 124 usecs from the "mainbang" to allow for the delay in the pulse compressor. One of the outputs is the "RANGE BUG", this output should be connected to one of the input channels on the A-scope. This bug is moved using the "RANGE BUG CONTROL" knob on the control panel. The other output is a 5 MHz clock which can be used as input to the range tracker allowing it to be operated without the system clock. (Schematics and timing diagrams are in appendix A.)

f. THE MICROPROCESSOR BOARD

The microprocessor board is a single board computer from Intel called the iSBC 80/24. It includes an Intel 8085A-2 microprocessor; 4K bytes of RAM; 8K, 16K or 32K bytes of ROM depending on whether 2716, 2732 or 2764 EPROM's are used (2716's were used in this project); six programmable 8 bit I/O ports; one programmable serial channel; a programmable interval timer and a programmable interrupt controller. The board has been expanded with two iSBX multimodule boards, one serial and one parallel, to increase the I/O capability to 72 bits of parallel I/O and two RS232 interfaces. Finally the iSBC 80/24 plugs into the Intel Multibus so that the board can communicate with the Timing-Control board and the Range Finder board. (Cabling for the microprocessor board is documented in appendix A.)



CONTROL & DATA FLOW GRAPH

FIGURE 9.

4. SOFTWARE

a. 8085 SOFTWARE

All 8085 software for this project was written totally in assembly language and was assembled using an 8080 cross assembler on the Signal Processing Lab's HP1000 computer. Also, a loader program was developed on the HP to allow the use of the LAB's PROM programmer for 8085 machine language.

The software in the 8085 accomplishes a number of tasks. First of all it implements an Alpha-Beta tracking filter and in conjunction with the hardware provides a range gate for the monopulse angle tracker. It provides the HP computer with range and tracking information to allow it to set up a recording window. It drives the S-band antenna to any azimuth given to it by either the L-band PPI or the HP computer. In conjunction with a program on the HP computer it provides 20 programmable gates for system timing. Finally, through a serial link to the Target ID lab's VAX computer , it provides target position data to ID programs.

The programs occupy about 2.5 K bytes of ROM and use less than 100 bytes of RAM for buffering and data basing. The software is modular in design for easy debugging and is broken down as indicated in the flowgraph in figure 9. The design of the software was broken down into 4 functions: the Alpha Beta tracker; the HP serial link (programmable gates); the Target ID/VAX link and those functions which require a lot of overhead and can be timeshared (called housekeeping functions). Each of these functions is comprised of a number of subfunctions or software routines which can be defined as either interrupt service routines or subprograms. The function of each of these routines is described in the following sections. A source listing of each routine is also provided in appendix B.

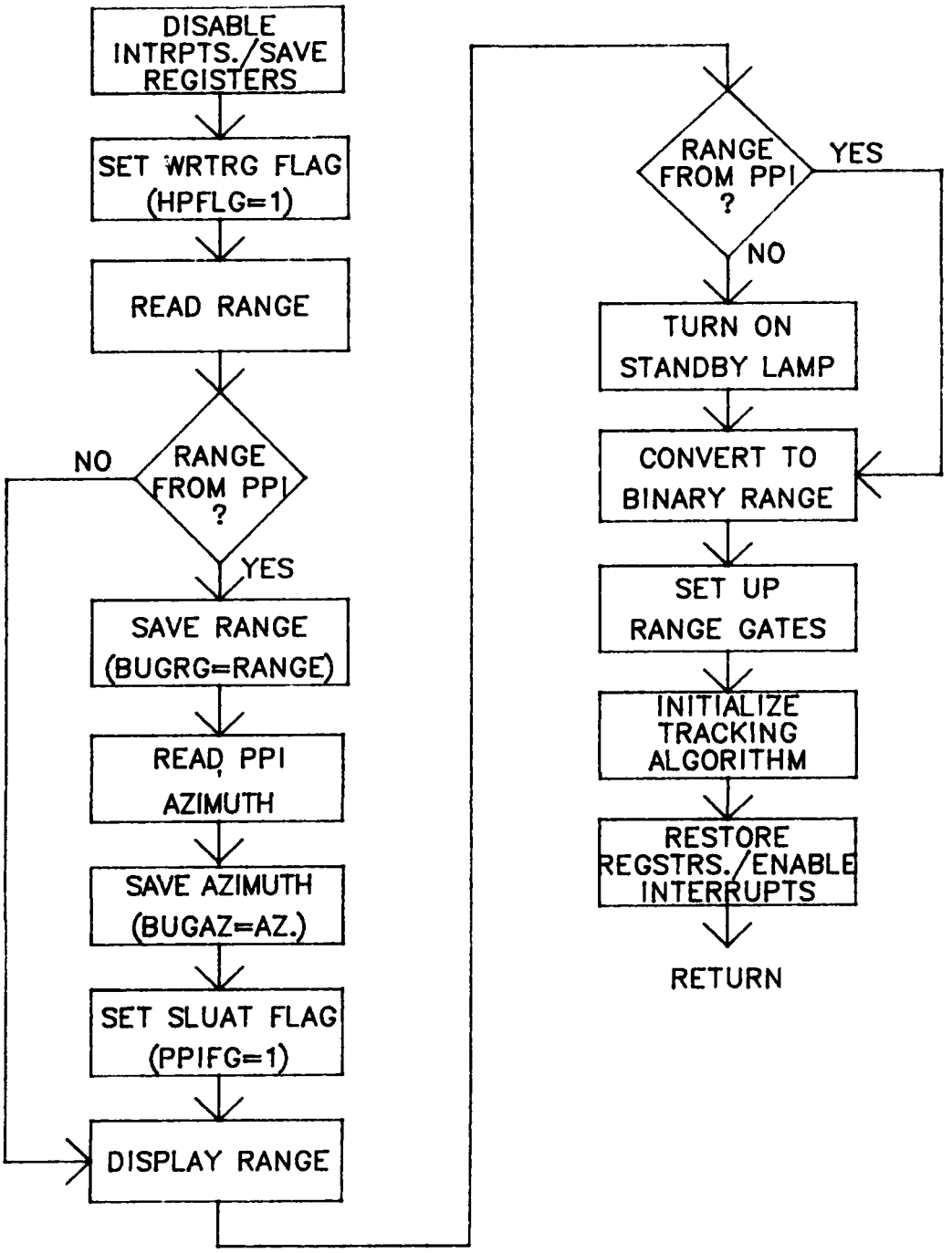
(1) THE ALPHA BETA TRACKER

The Alpha-Beta tracker consists of four interrupt service routines each of which handle a certain portion of the tracking algorithm. Each of these routines will be discussed in the logical order in which they are executed starting with tracker initialization and the IRINT routine.

(a) IRINT ROUTINE - TRACKER INITIALIZATION

The IRINT routine handles the initialization of the tracker. This involves accepting an initial range value from an external source and setting up the tracking gates at this range. It also involves initialization of tracking parameters and other flags and values that are used by the Alpha-Beta algorithm. The execution of this routine starts with reading an initial range from the initial range control board. Contained in this value is a flag used to determine the source of the range information; either the L-band PPI or the A-scope. If it is determined that the PPI is the source, the azimuth of the PPI cursor is also read and a flag is set (PPIFG=1) to trigger the execution of the SLUAT housekeeping routine. SLUAT will drive the S-band antenna to the proper azimuth.

Once the source of the range data is determined the range value is displayed on the front panel of the range tracker. The "STANDBY" indicator light is also turned on if the range information came from the A-scope, if not it will be turned on by the SLUAT routine once the antenna is in the correct position. The initial range value is then converted from BCD to binary for computational purposes. This conversion however does not go from BCD range in miles to binary range in miles. The binary value is actually the time delay in 0.2 usec increments from the PRF trigger to the indicated range. This value is used because it maps directly onto the 5 Mhz counters which are used to produce the range gates. IRINT takes this value and interfaces with the Timing-Control board to set up the two range gates. This routine sets a flag (HPFLG=1) which activates another housekeeping routine called



FLOW CHART – IRINT ROUTINE

FIGURE 10.

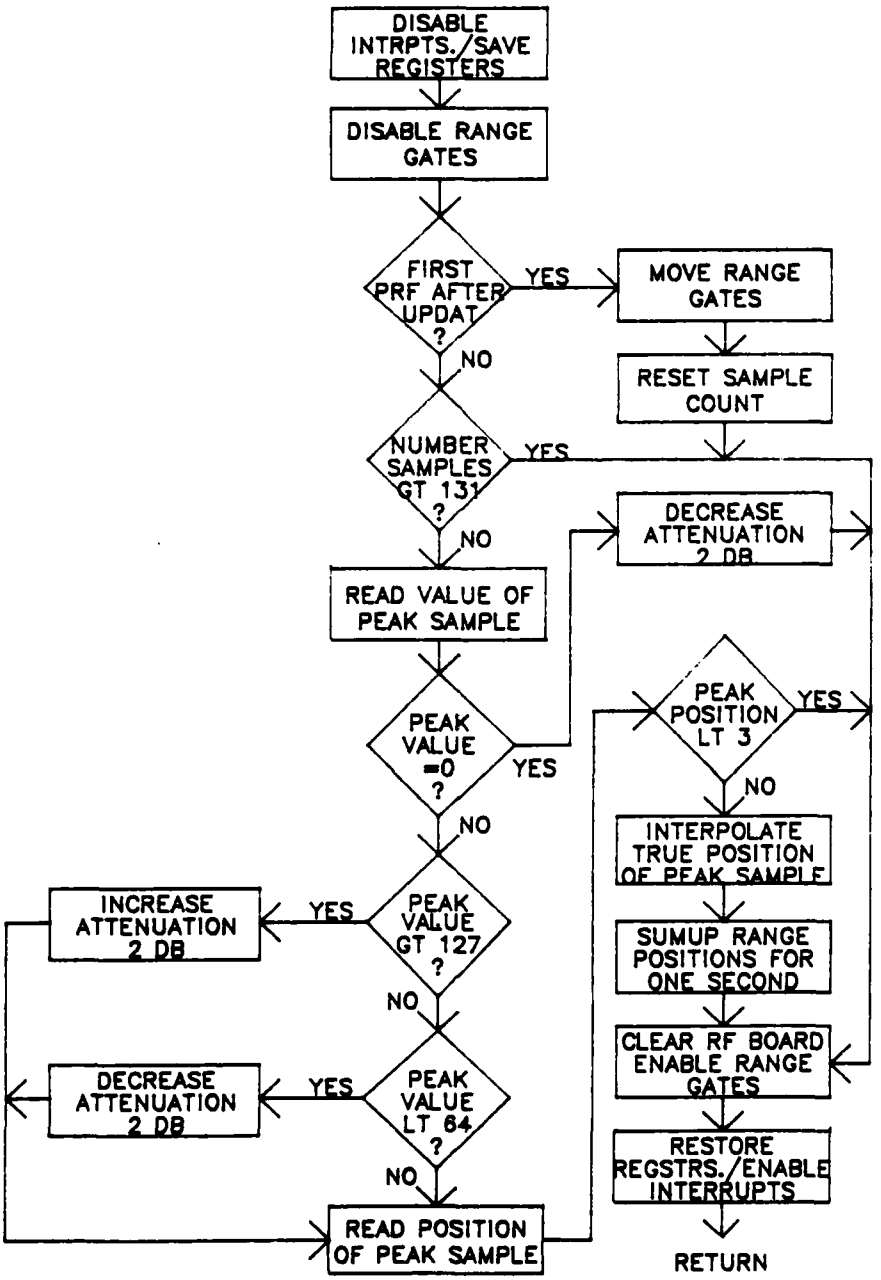
WRTRG. WRTRG will write the binary initial range value and the position of the PPI cursor to the HP computer. The IRINT routine is initiated by pressing the "INITIAL RANGE" button on the front panel of the range tracker. (A flow chart for IRINT is shown in figure 10.)

(b) EWINT ROUTINE - END OF RANGE WINDOW INTERRUPT

The EWINT routine implements four different functions: it interfaces with the Timing-Contol board to update the range gates; it applies an interpolation function to the data incoming from the Range Finder board to estimate the position of the target in the range window to 1/6 range cell (15 meters); it prefilters the data before it goes to the Alpha-Beta algorithm to weed out any bad data and to lower the data flow; it monitors the amplitude of the incoming data and provides a feedback for the gain control.

A flow chart for this routine is shown in figure 11. EWINT is initiated every PRI immediately following the tracking range gate. The first thing the routine does is determine whether the Alpha-Beta algorithm (UPDAT routine) has just finished executing. This is done by checking a flag which is set at the end of the UPDAT routine. If this flag is set then the tracking algorithm has updated the target position and the range gates should therefore be moved. The EWINT routine is used to do this because the new values can be written and the range gates re-enabled between PRF triggers. The UPDAT routine would not be satisfactory for doing this because it is not synchronous with the PRF trigger and chances are it would be updating the range gates at the same time they are triggered causing faulty operation. Once the EWINT routine has updated the range gates it immediately enables them, clears the range finder board and returns. All other functions of the routine are ignored to be sure that the gates are re-enabled in time for the next PRF trigger.

If it is determined that the gates need not be updated then the routine moves on to its other functions. The first test encountered checks to see how many valid range samples have been taken since the



FLOW CHART – EWINT ROUTINE
FIGURE 11.

last tracker update. Because of precision limitations the maximum number of samples that can be taken in the one second period between tracker updates is 131. Once this limit is reached all input data is ignored until after the next tracker update (UPDAT interrupt). The reasons for this limitation are given later in this section. If the number of samples is less than the limit then the Range Finder board is interrogated to get the peak value of the target and the values of the targets adjacent samples. The target peak is then tested to see if it is greater than zero. If the peak is zero then the target did not cross the threshold. This could be caused by target scintillation or a lost track. When this happens the routine decreases the attenuation of the incoming signal hoping to peak the target up above the threshold. The routine then resets the Range Finder board and returns.

If the target has crossed the threshold and the peak value is not zero then the value of the peak is tested to see if it lies within an acceptable range. If the peak is outside of the range then the attenuator is adjusted appropriately to either increase or decrease the value of the peak. This acceptance window lies between the values 63 and 128. This is checked by testing the 2 most significant bits in the 8 bit peak value. If the upper bit is a "1" then the attenuation is increased 2 dB to lower the peak. If the two upper bits are both "0" then the attenuation is decreased 2 dB to bring the peak up.

The next test encountered checks the validity of the position of the peak. The way in which the Range Finder board interprets a peak requires that at least 3 samples be taken from the A/D converter before a peak can be considered valid. If less than three samples are taken then the last (third) sample register is still cleared to zero causing an invalid comparison. This is illustrated by the Range Finder board's timing diagrams which are located in appendix A. If the position of the peak is found to have occurred before 3 samples were taken the data is rejected, the Range Finder board is reset and the routine returns.

At this point the input data is considered valid and the data manipulation begins. The first algorithm the data is subjected to is an interpolation routine which estimates the true position of the target in the tracking gate to 1/6 of a range cell (15 meters). The calculation uses the following equation :

$$X = -((S(i)-S(i+1))-(S(i)-S(i-1)) \;/\; 2*((S(i)-S(i+1))+(S(i)-S(i-1))$$

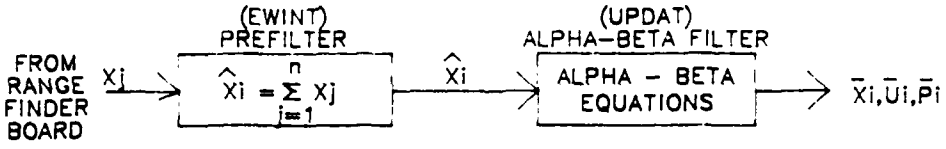
WHERE : X = ITERPOLATED POSITION OF THE PEAK RELATIVE TO SAMPLE i (PEAK SAMPLE)

S(n) = AMPLITUDE OF THE SIGNAL AT SAMPLE n

i = RANGE COUNT POSITION OF THE SAMPLE WITHIN THE RANGE GATE

Once X is found it is rounded off to the nearest 1/2. Once rounded the value of X is either -1/2, 0, or +1/2 corresponding respectively to: a reduction in the measured position of 1/2 a sampling period, no change in the measured position and an increase in the measured position of 1/2 a sampling period. Since sampling is done at a 5 MHz rate the sampling period is 0.2 usec and the measured position is in 0.2 usec increments. Since all calculations are done using integer values, half of a sampling period could not be added or subtracted from the measured position without roundoff error. To get around this problem the value of the measured position and the value of "X" were doubled and then added together thus giving a 0.1 usec or 1/6 range cell resolution.

The last function of this routine is to compress the Range Finder data in order to reduce the data flow into the Alpha Beta algorithm. The data that is received every PRI from the Range Finder board, after going through validity checks and the interpolation logic, is summed up over the one second period between tracker updates. A record of the



EWINT – UPDAT RELATIONSHIP
FIGURE 12.

TABLE 1.
ALPHA – BETA EQUATIONS

$$\bar{X}i = A*\mathring{X}i+(1-A)*(\bar{X}(i-1)+\bar{U}(i-1)) = A*\mathring{X}i+(1-A)*\bar{P}(i-1)$$

$$\bar{U}i = B*(\mathring{X}i-\bar{X}(i-1))+(1-B)*\bar{U}(i-1) = B*(\mathring{X}i-\bar{P}(i-1))+\bar{U}(i-1)$$

$$\bar{P}i = \bar{X}i+\bar{U}i = (A+B)*\mathring{X}i+(1-A-B)*\bar{P}(i-1)+\bar{U}(i-1)$$

$$\mathring{X}i = \hat{X}i / n$$

WHERE:

Xj := POSITION OF TARGET IN RANGE GATE AT SAMPLE j

$\hat{X}i$:= SUM OF ALL SAMPLES (Xj) TAKEN OVER A ONE SECOND PERIOD

$\mathring{X}i$:= AVERAGE POSITION OF TARGET IN TRACKING GATE FROM i–1 to i

n := NUMBER OF SAMPLES (Xj) SUMMED OVER A ONE SECOND PERIOD

$\bar{X}i$:= ESTIMATED TARGET POSITION AT TIME i BASED ON OBSERVED DATA UP TO AND INCLUDING $\mathring{X}i$

$\bar{U}i$:= ESTIMATED TARGET VELOCITY MULTIPLIED BY TIME BETWEEN UPDATES

$\bar{P}i$:= EXTRAPOLATED TARGET POSITION FOR TIME i+1 BASED ON OBSERVED DATA UP TO AND INCLUDING $\mathring{X}i$

A := THE VALUE OF ALPHA (1 IF DURING INITIALIZATION, 1/2 OTHERWISE)

B := THE VALUE OF BETA (0 OR 1 DURING INITIALIZATION, 1/6 OTHERWISE)

number of samples added is also kept to allow the UPDAT routine to take an average which can then be used in the Alpha Beta algorithm. This final summing function limits the number of samples which can be taken. The widest tracking gate used is 50 usec wide (4.2 NM). If a target were found to be at the very end of this gate the position value of the target within the tracking gate using 0.1 usec resolution would be 500. Since the summing register is only 16 bits wide this limits the number of samples added to : 65535 / 500 = 131 samples.

At the end of the one second summing period (the UPDAT interrupt occurs) the UPDAT routine will use the position summation and the number of samples to calculate an average position of the target within the tracking gate. This position value is then added to the range position of the tracking gate to obtain a true range to the target which can then be used in the Alpha Beta tracking equations.

By averaging any hardware inconsistencies that may have entered into the range measurement on a PRI basis will be cancelled. Also, by updating the tracking filter on a one second basis, the velocity estimate of the target will be less effected by roundoff error as compared to an update every PRI, especially for very slow moving targets. The relationship between the EWINT and UPDAT routines is illustrated in figure 12.

(c) UPDAT ROUTINE - ALPHA BETA FILTER

This routine implements the Alpha-Beta tracking equations shown in table 1 as well as some filtering techniques to guard against range gate capture by stronger targets or clutter which may enter into the tracking gate. This routine is executed on a one second basis. The EWINT (prefilter) routine collects range measurements over the one second period and sums them to provide an input to the Alpha-Beta filter. Referring to table 1, the X(j) are the inputs from the Range Finder board, the X(i) are the sum of the X(j) data over a one second period and n the number of inputs summed. The Alpha-Beta filter produces X(i) which is the average of the X(j) data and X(i), U(i) and P(i) which are

estimates of current target position (X(i)), target velocity multiplied by the time between estimates (U(i)) and the predicted range of the target for the next estimate (P(i)).

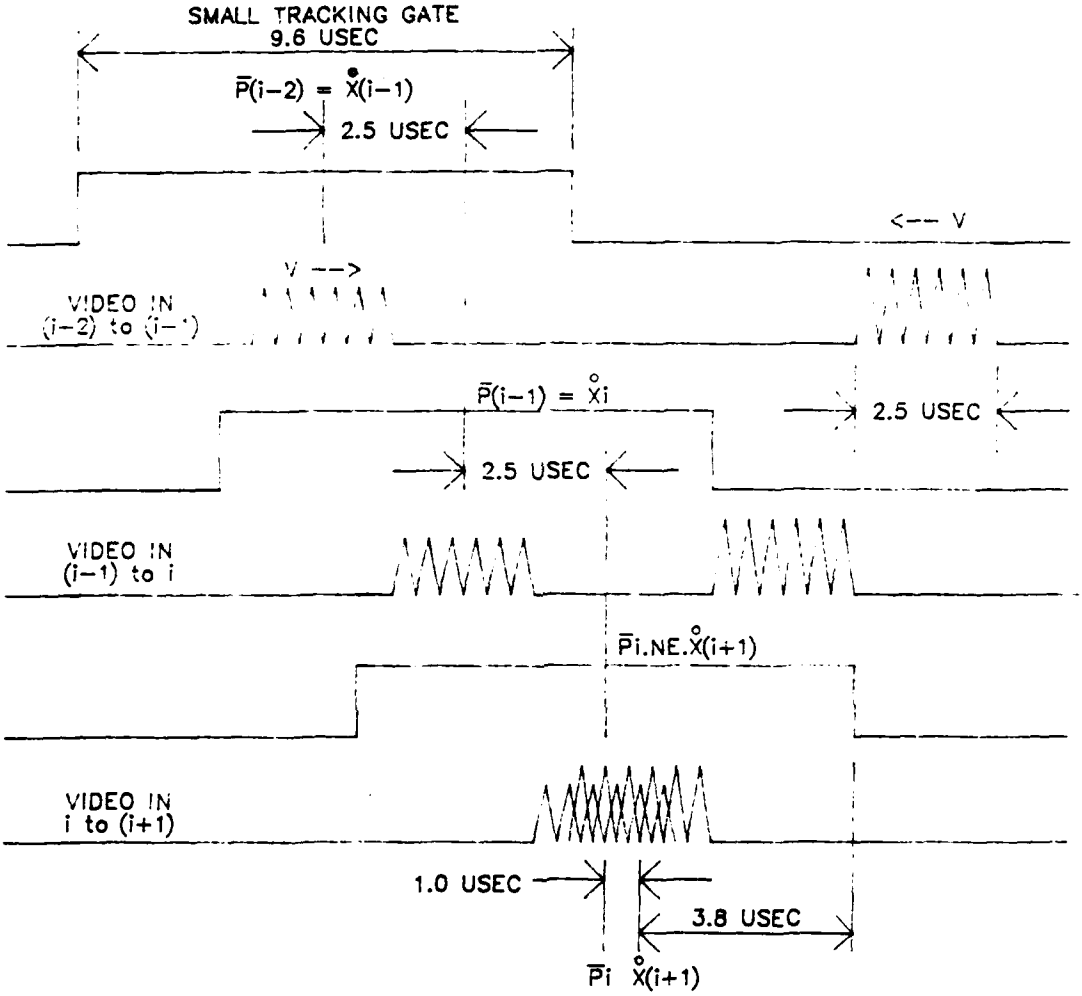
The Alpha-Beta filter is initialized by making Alpha=1 and Beta=0 for the first X(i) and Alpha=1 and Beta=1 for the second X(i). Therefore it takes two seconds for the filter to initialize. During this time the wide detection window gate (4.2 nm) is used. Once the filter is initialized, Alpha=1/2 and Beta=1/6 and the detection window gate will be 4.2 nm, 1.6 nm or 0.8 nm depending on the error in the tracking algorithm. The error is the difference between the measured position (X(i)) and the predicted position from the last estimate (P(i-1)). The values of Alpha and Beta, the detection window gate sizes and the various filtering techniques are based on targets with a maximum velocity of MACH 1 and a maximum turn acceleration of 5 G's.

Through multiple iterations of the tracking algorithm it was discovered that the tracker performance is based on the change of velocity of the target and on the smoothness of that change. It can be shown that the maximum error in the tracker occurs just after a transition from maximum acceleration to maximum deceleration. A series of iterations were done on the tracker using the stated values of Alpha and Beta and the velocity and acceleration limits. The tracker was subjected to an acceleration of 5 G's over several iterations until the simulated target had reached the maximum velocity of MACH 1. At this point the tracker was subjected to a deceleration of 5 G's for several more iterations and the performance of the tracker was observed. At no time during this simulation did the absolute error in the tracker ever exceed 1.8 usec or the change in error from one update to the next ever exceed 0.4 usec. The absolute error, as stated before, is the difference between the predicted and the measured position of the target. As long as the target being observed does not exceed the stated performance limitations then the error in the tracker should not exceed these observed values. These values are therefore used in the tracker as error limitations. If these limits are exceeded the tracker assumes

some type of interference in the tracking gate. This interference could be caused by another aircraft, chaff, clutter, etc. that may have moved into the tracking gate. When this occurs the tracker enters into "COASTING" mode. In this mode the returns from the prefilter are ignored and the tracker will move the range gates according to the last velocity and acceleration measurements. This prevents range gate capture by stronger returns that may enter into the range gate. The gate used during "coasting" is the wide, 50 usec, gate that is used during tracker initialization. Before pressing the "COAST UNLOCK" button to get out of this mode the operator should be sure that no interference is present in this wide gate that is "brighter" than the target in question. Upon leaving this mode the tracker goes into a re-initialization to acquire up-to-date tracking information.

The width of the gates and their use is also governed by the velocity and acceleration limitations. It was found through simulation that a target moving at a constant velocity will never impose an error in the tracker greater than 0.2 usec after tracker initialization. This value, doubled, is used to govern the use to the small (9.6 usec) tracking gate. This gate is used if the observed error in the track is 0.4 usec or less, corresponding to a target that is experiencing very little or no acceleration. It may be asked: If this gate is only tracking targets with 0.4 usec error, why is the gate 9.6 usec wide?. The extra width of the gate is used to detect any interference moving into the gate. Over the one second update period a target flying at MACH 1 would move 2.5 usec in range, this then limits the movement of the range gate on every update to 2.5 usec. This velocity limit also affects any other targets which may be moving into the range gate. Therefore the worst case in detecting a target moving into the range gate would occur when 2 targets are closing on each other at MACH 2 speed. This worst case scenario was used in designing the gate widths.

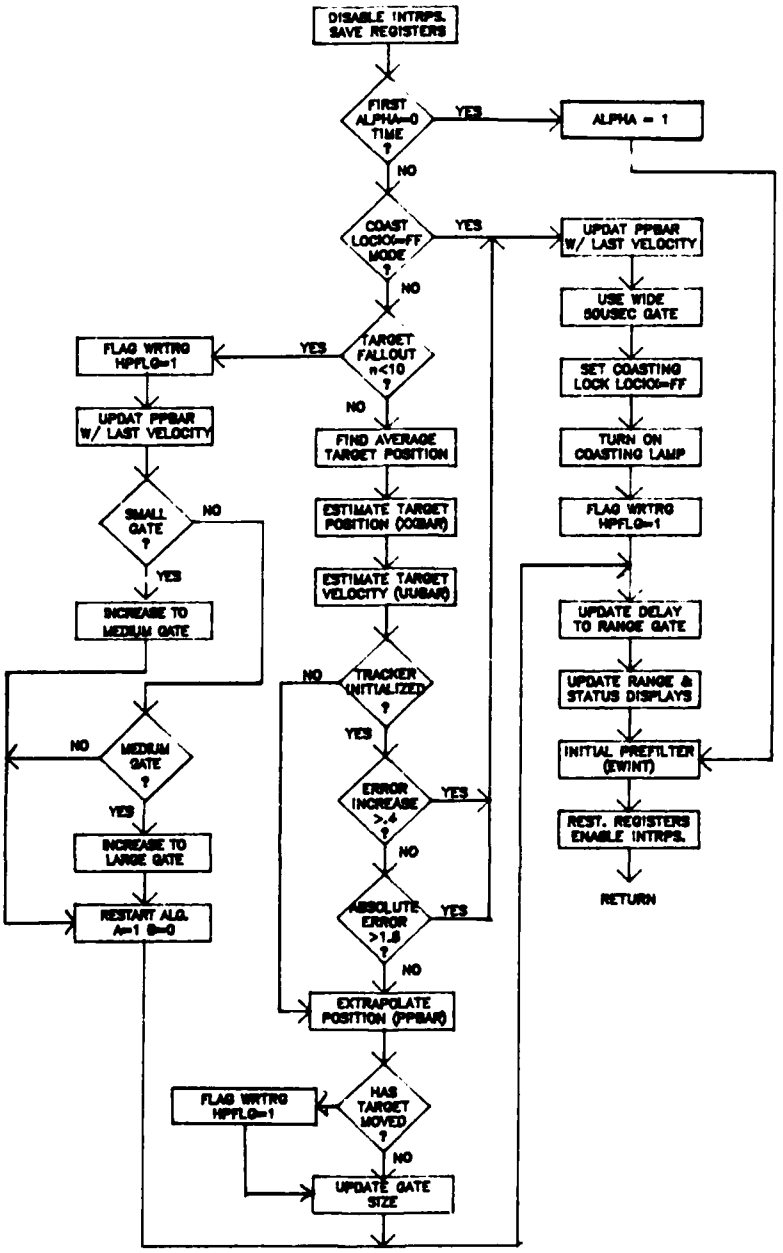
The small, 9.6 usec range gate was designed so that if a brighter target enters the gate it will cause an error increase that exceeds the 0.4 usec error increase limitation over a one second period, therefore



INTERFERING TARGET MOVING INTO
TRACKING GATE (WORST CASE)
FIGURE 13.

forcing the tracker to coast. Referring to the example in figure 13 a target is being tracked with the small tracking gate, the tracker then is experiencing an error of 0.4 usec or less. The velocity of the target being tracked as well as the velocity of the incoming interfering target is MACH 1 therefore representing the worst case scenario. Since the maximum error in the tracker is 0.4 usec and the maximum allowable increase in error between updates is 0.4 usec, the absolute maximum error that can be tolerated at X(i+1)-P(i) is 0.8 usec. As can be seen the worst case happens when the position of the interference corresponds with the gate edge at the end of the interval between P(i-1) and P(i), just before the target enters the gate. When the gate is updated it is moved 2.5 usec in the direction of the incoming target, therefore moving the interfering target 2.5 usec into the tracking gate. Over the one second period that P(i) exists the target will have moved another 2.5 usec towards the center of the gate. If this interfering target is of greater amplitude than the one being tracked the tracker will mistake it for the target being tracked and will compute its average position within the tracking gate. In this worst case example that position will be 3.8 usec from the gate edge from which the target entered. This position will give a minimum increase in error of 0.6 usec and a minimum absolute error of 1 usec thus causing the tracker to coast. It is also possible that the absolute error in the track could be greater than the 1.8 usec error limit if the relative velocity of the 2 targets was low enough. This gate width could have been designed so that an error greater than the 1.8 usec limit would occur for the worst case example but the gate would have to be wider and the performance would not be any better. By studying the example in figure 13 it can be seen how a minimum gate size of 9.6 usec was obtained.

The medium size gate (19.2 usec) is used by the tracker if the error in the track is greater than 0.4 usec and no more than 1.8 usec. This error acceptance window will handle any targets that are maneuvering within the specified limitations. The width of this gate is governed in much the same way as the smaller gate. If the minimum gate width is computed using 1.8 usec as the maximum tolerable error instead

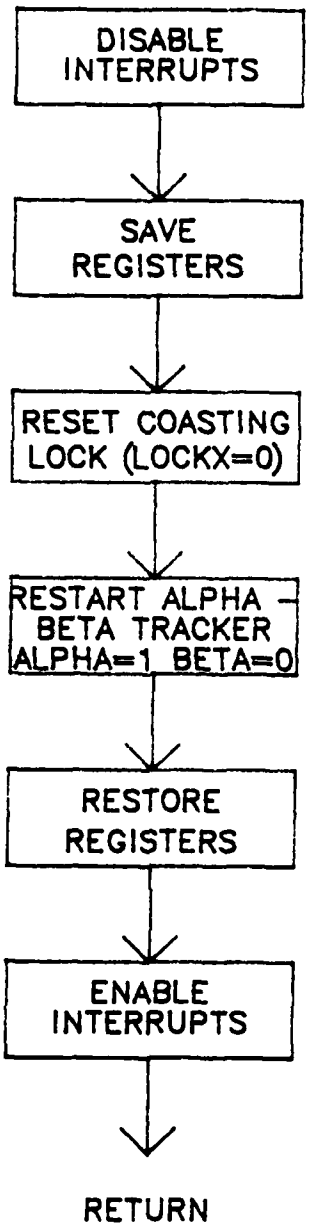


FLOW CHART – UPDAT ROUTINE
FIGURE 14.

of the 0.8 usec used for the small gate a minimum width of 11.6 usec is obtained. This is the absolute minimum width needed by the medium size gate to handle the MACH 2 closing velocity. However, as a go between for the small gate and large gate (50 usec), the width of the medium size gate was arbitrarily increased to 19.6 usec.

The largest gate used by the tracker is 50 usec wide. This gate is used during tracker initialization, coasting and also after coasting to reacquire the target. The width of this gate is limited by the 8-bit range window counter located on the range finder board. This counter keeps track of the position within the range window as the logic searches for the highest peak. It is incremented at a 5 MHz rate which at the end of a 50 usec gate has reached a value of 250.

A flow chart for the UPDAT routine is shown in figure 14. The first test in this routine checks to see if UPDAT has been executed yet. The first time this routine is initiated after the IRINT routine it merely sets Alpha=1 and returns. This ensures that the data for the first tracker update was accumulated over a full second. Next, the coasting lock is checked to see if the tracker is in coasting mode, if it is the tracker is updated according to the last known velocity of the target and the routine returns. If the tracker is not in coast mode the number of samples integrated by the EWINT routine is checked to determine the stability of the data. If the number of samples integrated by EWINT is less than ten the routine rejects the data with the reasoning that the target is too unstable to establish a good track. After these three checks are passed the inputs from the EWINT routine are averaged, the tracking equations are updated and if the tracker is initialized (Alpha=1/2, Beta=1/6) the error in the track is determined and the coasting check is made. After the predicted position is computed (PPBAR) a check is made to see if the target has moved any since the last update. If it has HPFLG is set to "1" which will cause the execution of the WRTRG routine. This routine will write the new range estimate to the HP computer. HPFLG is also set during coasting mode and when the the track is extended because of target fallout. The



FLOW CHART – UNLOK ROUTINE
FIGURE 15.

gate width used is then updated based on the error in the track. If the error is less than or equal to 0.4 usec the small gate is used, if it is greater than this the medium gate is used or if the tracker is coasting or is uninitialized the large gate is used. The next function in this routine computes the delay to the start of the range gate so that the EWINT routine only has to load the value and store it in the programmable gate. The routine then converts the binary range representation (PPBAR) into BCD range in nautical miles and displays it and the tracking status on the front panel. The prefilter is then reset and the routine returns.

The UPDAT and EWINT interrupts are both initiated internally. On power up their interrupting mechanism is disabled and is not enabled until the angle track locked signal is received.

(d) UNLOK INTERRUPT - COASTING UNLOCK ROUTINE

The last software routine included in the Alpha-Beta tracker functions is the UNLOK interrupt. This is a very short routine that takes the tracker out of coasting mode. It resets the flag which locks the tracker in coasting mode (LOCKX) and it resets Alpha=1, Beta=0 so that the tracker will re-initialize. This routine is initiated by pressing the "COAST UNLOCK" button on the front panel of the tracker. Taking the tracker out of coasting mode causes it to lock up on the largest return in the tracking gate. The operator should therefore be sure that any clutter present in the range gate during coast unlock is weaker than the target in question. A flow chart for this routine is shown in figure 15.

(2) HP SERIAL LINK - DOWNLOADER

The HP serial link contains one routine called HPINT. This routine is initiated when a character is received from the HP computer. The value of this character determines what function HPINT will implement. If the character received is a "20" hex the routine will download from the HP all the information necessary for programming a specific

TABLE 2.
INPUT FORMAT FROM HP – PROGRAMMABLE GATES

SEQUENCE OF ARRIVAL:	G 1	G 2	D 3	P 4	M 5	M 6	L 7	L 8	H 9	H 10	E 11	A 12

BYTE NUMBER

1 – HIGH ORDER BYTE OF THE ADDRESS OF THE CHIP WHICH CONTAINS THE GATE TO BE PROGRAMMED (ALWAYS 20 HEX).

2 – LOW ORDER BYTE OF CHIP ADDRESS, (COMMAND REGISTER ADDRESS BIT0 = 1)

3 – DISARM COMMAND FOR GATE TO BE PROGRAMMED, GATE MUST BE DISARMED BEFORE PROGRAMMING. THIS BYTE IS WRITTEN TO THE COMMAND REGISTER.

4 – SET DATA POINTER COMMAND, THE DATA POINTER MUST BE SET TO ACCESS THE DATA REGISTERS (MODE,LOAD&HOLD) ASSOCIATED WITH THE GATE TO BE PROGRAMMED. THIS BYTE IS WRITTEN TO THE COMMAND REGISTER.

5 – ONCE THE POINTER IS SET THE DATA REGISTERS ARE LOADED. THIS IS THE LOW ORDER BYTE OF THE COUNTER MODE DATA. IT IS WRITTEN TO THE DATA REGISTER (COMMAND REGISTER – 1).

6 – HIGH ORDER BYTE OF THE COUNTER MODE DATA. ALSO WRITTEN TO THE DATA REGISTER.

7 – LOW ORDER BYTE OF THE LOAD REGISTER DATA. WRITTEN TO THE DATA REGISTER.

8 – HIGH ORDER BYTE OF LOAD REGISTER DATA.

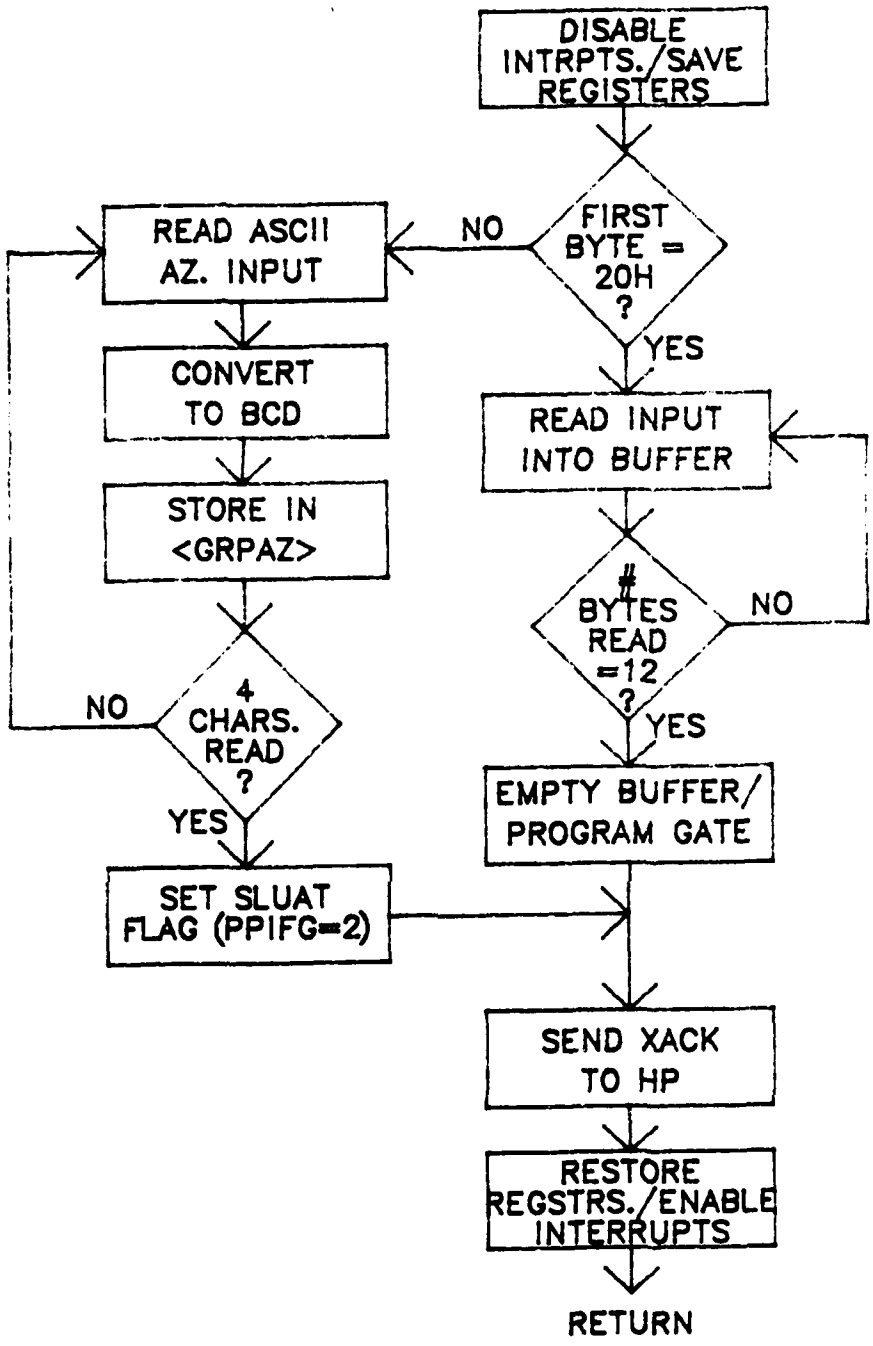
9 – LOW ORDER BYTE OF HOLD REGISTER DATA. WRITTEN TO THE DATA REGISTER.

10 – HIGH ORDER BYTE OF HOLD REGISTER DATA.

11 – OUTPUT CONTROL COMMAND. USED TO CONTROL HIGH/LOW ENABLE FOR COUNTER OUTPUT. WRITTEN TO THE COMMAND REGISTER.

12 – LOAD & ARM COUNTERS COMMAND – LOADS DOWNCOUNTER WITH VALUE FOUND IN LOAD REGISTER AND ENABLES OPERATION.

(NOTE: FOR FURTHER EXPLANATION ON THE USE OF THE COMMANDS AND DATA REGISTERS, SEE THE AM9513 SYSTEM TIMNG CONTROLLER USER'S MANUAL.)



FLOW CHART – HPINT ROUTINE
FIGURE 16.

programmable gate. The routine will download 12 bytes of information in the sequence shown in table 2, the first byte being the "20" hex which is the same for all twenty programmable gates. This first byte represents the high order byte of the memory location of the chip that contains the gate to be programmed. The meaning of each of the successive bytes and the register to which they are written are listed in table 2. Each of the bytes 3 thru 12 are written in the same sequence that they are received from the HP.

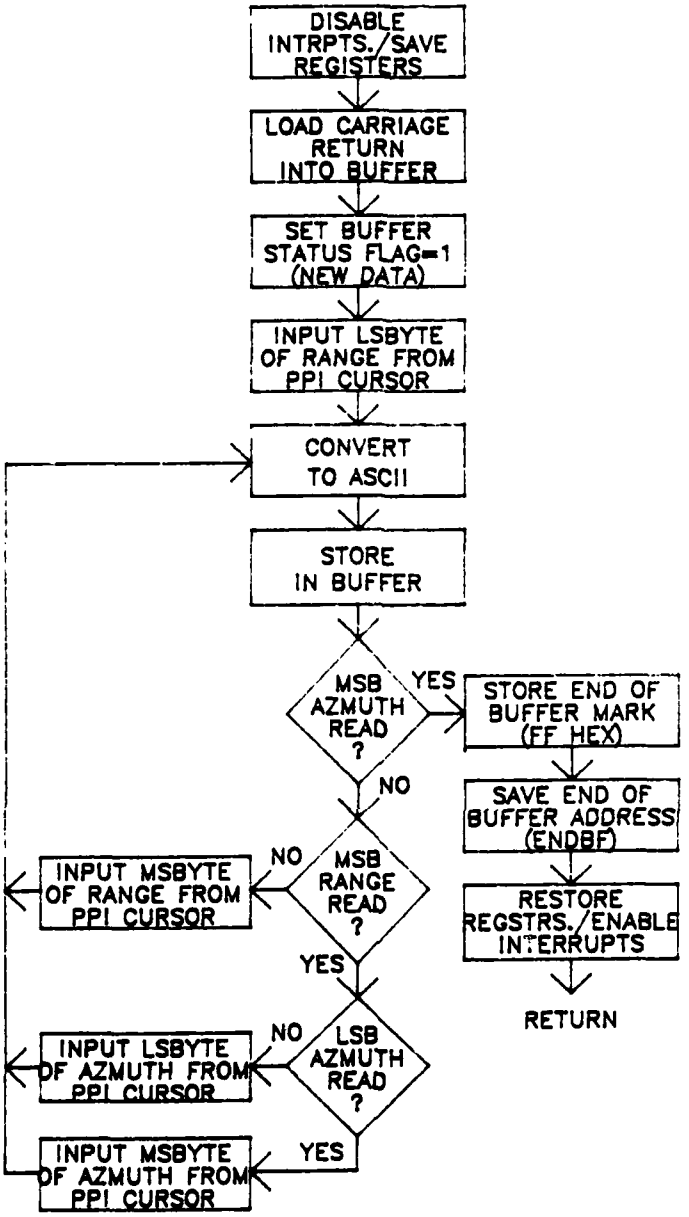
If the first byte received from the HP is not a "20" hex then this routine will download 4 ASCII characters from the HP. These characters represent a BCD azimuth position to which the S-band antenna is to be driven. These characters are downloaded starting with the most significant digit (hundreds of degrees) into a buffer. This azimuth value is converted into BCD and is stored in memory location "GRPAZ" (two bytes) for use later by the SLUAT routine. After conversion the flag PPIFG is set equal to "2", this will cause the execution of the SLUAT routine which will drive the S-band antenna to the azimuth position found in GRPAZ.

Before this routine returns it acknowledges the data transfer from the HP by sending a character back. This sychronization is needed to prevent the HP from overwriting previously transmitted data. A simplified flow chart is shown in figure 16.

(3) TARGET ID LAB/ VAX INTERFACE

This interface provides the Target ID Lab's VAX computer with range and azimuth information from the L-band PPI. The information is needed by ID programs to establish a track on the target to be identified. The interface is handled with two interrupt routines, one to handle the PPI input and the conversion of this input to ASCII (PPIIT) and one to drive the VAX serial interface (VAXIT).

A flow chart for the PPIIT routine is shown in figure 17. The routine is initiated by pressing the "TARGET ID" button mounted on the



FLOW CHART – PPIIT ROUTINE
FIGURE 17.

TABLE 3.
VAX BUFFER FORMAT

<-- WRITE

E	A	A	A	A	R	R	R	R	F	CR
1	2	3	4	5	6	7	8	9	10	11

READ -->

1 – END OF BUFFER MARK, LAST BYTE WRITTEN BY PPIIT.

2 – ASCII CHARACTER REPRESENTING THE MOST SIGNIFICANT DIGIT OF THE BCD AZIMUTH VALUE (IN DEGREES).

3 – TENS OF DEGREES (ASCII).

4 – ONES POSITION – AZIMUTH (ASCII).

5 – TENTHS OF DEGREES (ASCII).

6 – ASCII CHARACTER REPRESENTING THE HUNDREDS OF NAUTICAL MILES POSITON OF THE BCD RANGE VALUE.

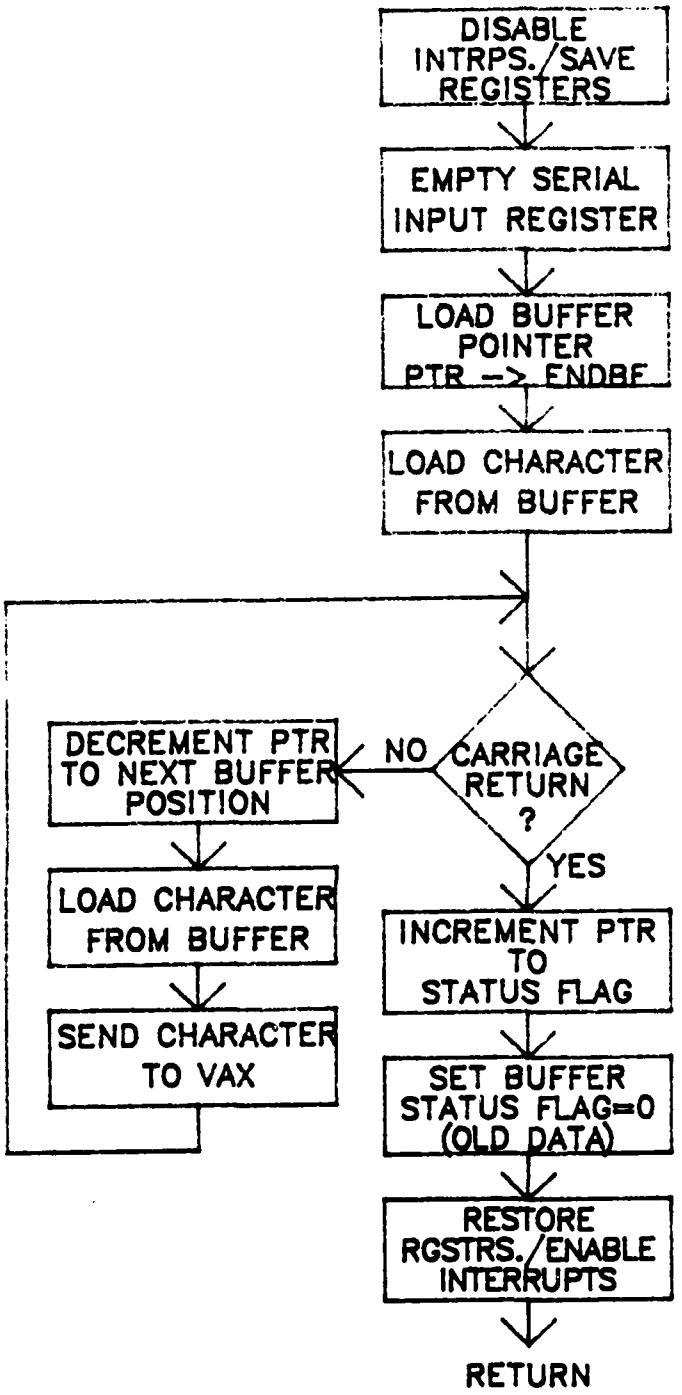
7 – TENS OF MILES (ASCII).

8 – ONES POSITION – RANGE (ASCII).

9 – TENTHS OF MILES (ASCII).

10 – BUFFER STATUS FLAG, "1" = NEW DATA, "0" = OLD DATA (ASCII)

11 – VAX TERMINATION CHARACTER, ASCII CARRIAGE RETURN.



FLOW CHART – VAXIT ROUTINE
FIGURE 18.

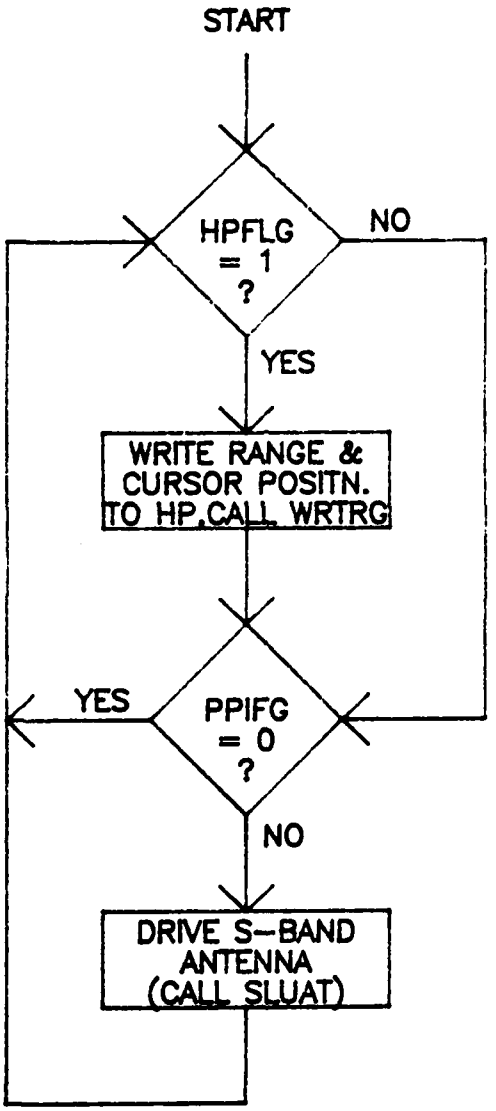
L-band PPI. This routine reads the azimuth and range of the PPI cursor, converts it to ASCII and sets up a buffer in the form shown in table 3. The buffer is filled starting with a carriage return (byte 11) and ending with an "end of buffer" marker (byte 1) which is used by the VAXIT routine as a delimiter.

This buffer is sent to the VAX in the opposite direction that it is filled (byte 1 to byte 11) by the VAXIT routine. A flow chart is shown in figure 18. Execution is initiated on receiving a character from the VAX. It first empties the micro's input register then proceeds to transfer the buffer. After the termination character is sent (carriage return) the buffer status is set to "0" indicating that the present buffer has been sent.

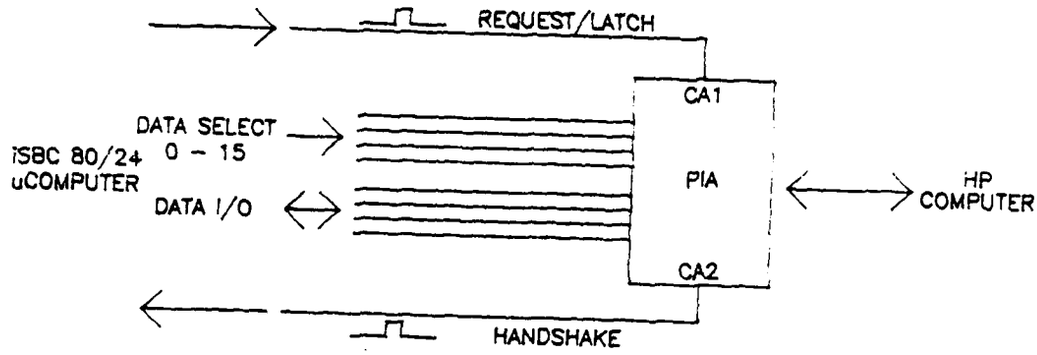
(4) HOUSEKEEPING FUNCTIONS

The housekeeping functions are those software routines in the tracker which require a lot of overhead and can be timeshared. Two such functions exist in the tracker: the SLUAT routine which is responsible for driving the S-band antenna and the WRTRG routine which reports target range to the HP computer. These routines are timeshared in the sense that if an interrupt occurs during their execution they will be suspended until the interrupt service routine has finished processing. They are in effect the lowest priority routines that are executed.

Referring to the Control and Data Flow graph in figure 9 the execution of these routines is initiated during the execution of a piece of software called ILOOP. This is merely an infinite loop that continually monitors the flags which govern the execution of the SLUAT and WRTRG routines. As seen in the flow chart in figure 19, ILOOP is executed until one of these flags is found to be set or, because the interrupts are not disabled, until an interrupt occurs. As described in previous sections the value of HPFLG and PPIFG are altered by various interrupts. HPFLG is set to "1" by the IRINT routine after receiving an initial range value and also by the UPDAT routine after the range estimate of the target is updated by the tracking algorithm. HPFLG can



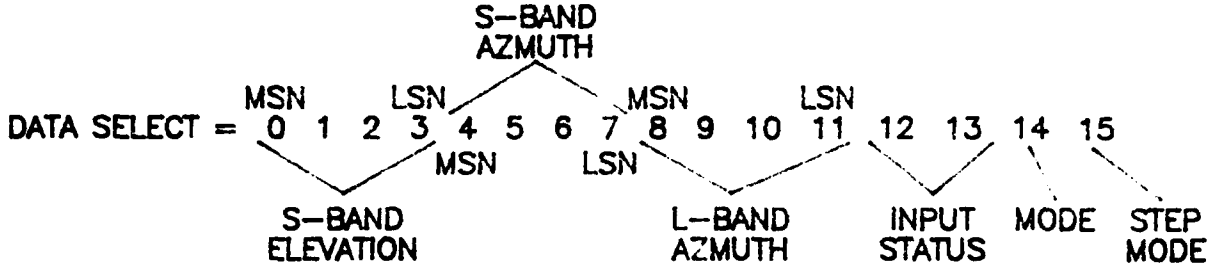
FLOW CHART – ILOOP
FIGURE 19.



HP PARALLEL INTERFACE
FIGURE 20.

TABLE 4.
PIA OPERATING MODES

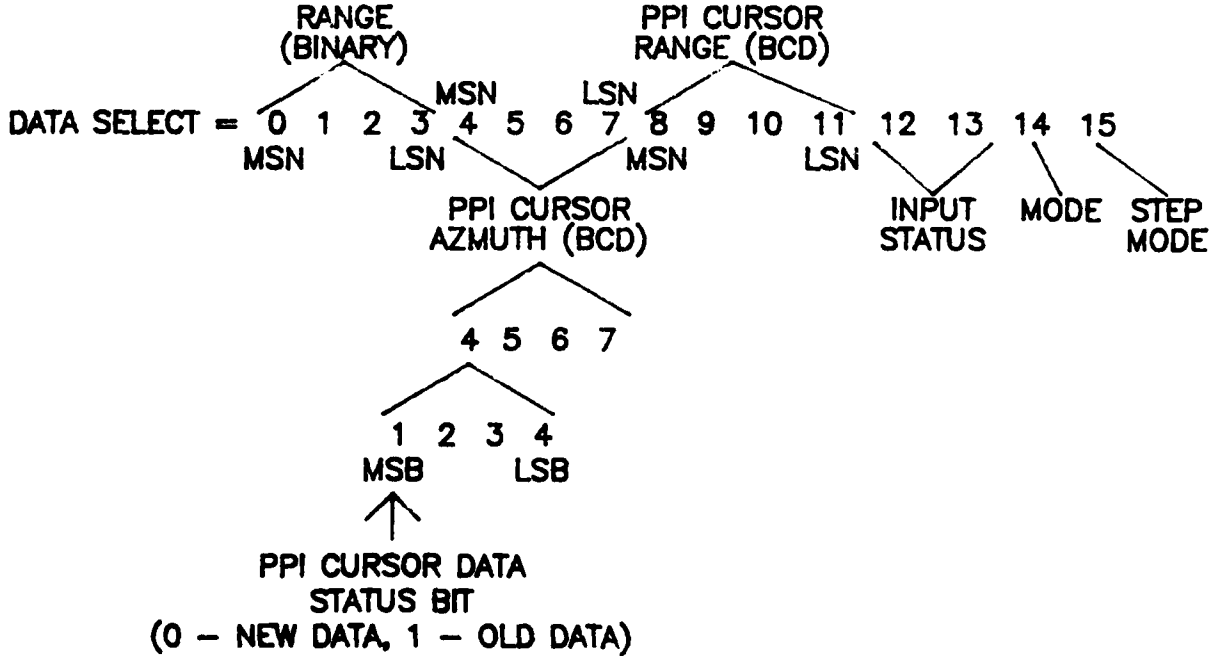
MODE 0 – BCD DATA – INPUT FROM HP



MODE 1 – BINARY DATA – (0–359.9) – INPUT FROM HP
DATA SELECT SAME AS MODE 0

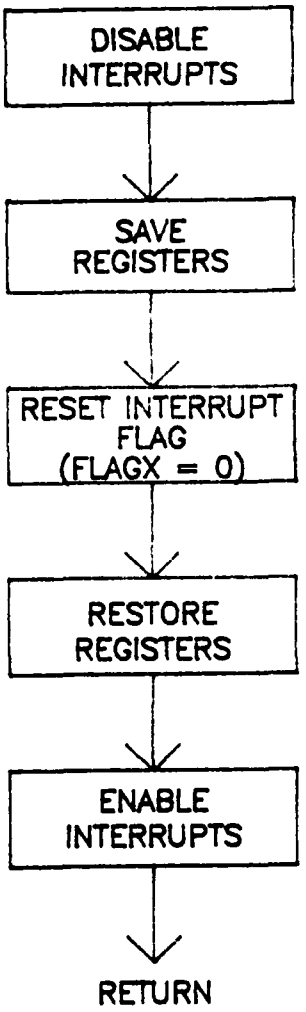
MODE 2 – BINARY DATA – (0–16383) – INPUT FROM HP
DATA SELECT SAME AS MODE 0

MODE 3 – RANGE & CURSOR POSITION OUTPUT TO HP

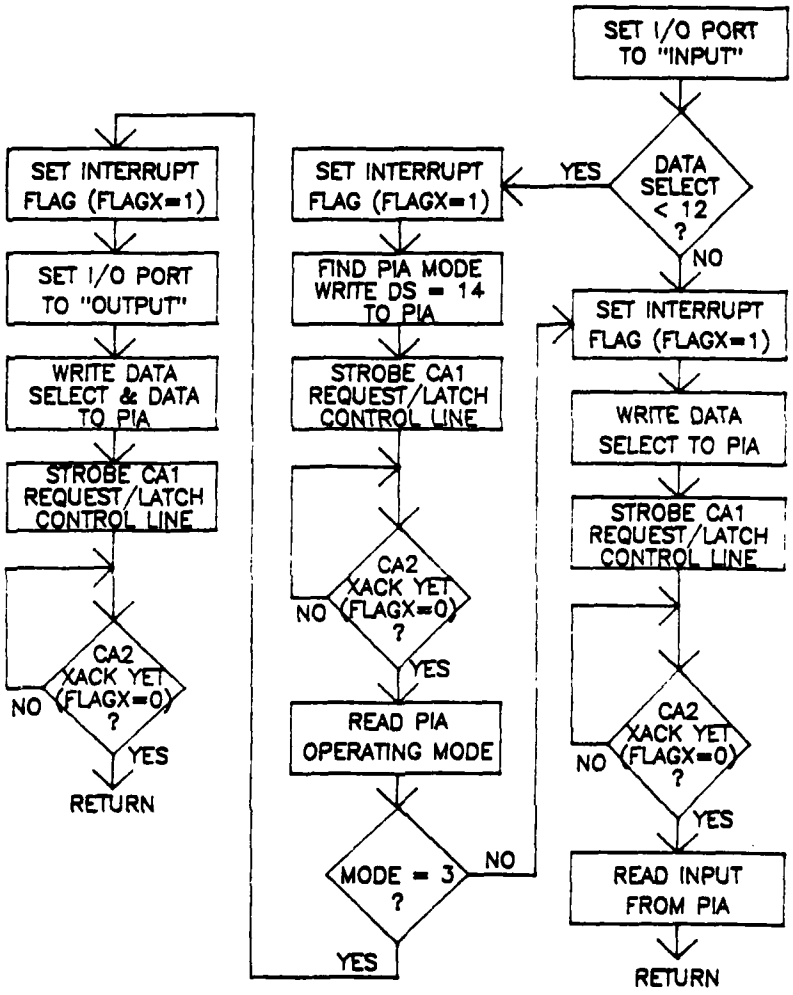


only take on two values either "0" or "1". PPIFG is also altered by two routines, however it may take on three values "0", "1" or "2". The IRINT routine will set PPIFG to "1" if the initial range is taken from the L-band PPI; this causes the S-band antenna to be driven to the azimuth of the PPI cursor which is saved in memory location BUGAZ. The HPINT routine will set PPIFG="2" if the HP computer downloads an azimuth position over the serial link. This azimuth position is stored in memory location GRPAZ.

These two housekeeping functions utilize two subroutines (RETRV and PIAIO) and an interrupt service routine (PIAIT) to communicate with the HP parallel interface. This interface provides the micro with the azimuth and elevation of the S-band antenna as well as the azimuth of the L-band antenna in two binary forms and in BCD. The interface also provides a means of transferring range information and PPI cursor information to the HP computer. Referring to figure 20 this interface consists of a peripheral interface adaptor (PIA) that is programmed to operate in the four different modes seen in table 4. All data transfers are initiated by the microprocessor by setting the proper data select lines and strobing the control line CA1. The PIA responds by either outputting or inputting data, depending on the mode in which it's operating, and strobing the CA2 handshake line. The interface is operated by first putting the PIA in the proper operating mode. First, the mode of the PIA is read, this is accomplished by setting the data select equal to "14" and strobing the CA1 line (regardless of what mode the PIA is in). This causes the PIA to output its current operating mode (0, 1, 2 or 3) on the data lines and to strobe CA2. If the PIA is not in the correct mode the above sequence is repeated using data select = "15" until the proper mode is reached. Data Select = "15" causes the mode to be stepped and the new mode to be output on the data lines. Once the PIA reaches mode 3 the next mode step will be mode 0. Once the PIA is in the proper mode data can be sent/received one nibble (4 bits) at a time. A data transfer is accomplished in the same fashion as reading the mode. The proper data select lines are set corresponding to the nibble wanted and the CA1 line is strobed. Depending on the mode of



FLOW CHART – PIAIT ROUTINE
FIGURE 21.



FLOW CHART – PIAIO ROUTINE
FIGURE 22.

operation, one nibble will be received from the HP (modes 0, 1 and 2) or sent to the HP (mode 3). In the case of receiving information, the data lines are not read until the CA2 handshake is received. During mode 3, the data select and data lines are set before the CA1 strobe is given. These lines must remain stable until after the CA2 handshake is received, verifying the transfer. Also shown in table 4 is the status flag for the PPI cursor azimuth and range information for mode 3. The most significant bit of nibble 4 is a flag which allows the HP to determine the status of the cursor information. If this bit is a "0", new data appears in nibbles 4 thru 11; if it is a "1" then previously transferred data remains in these positions. Nibbles 0 thru 3 are always new everytime this data is transferred.

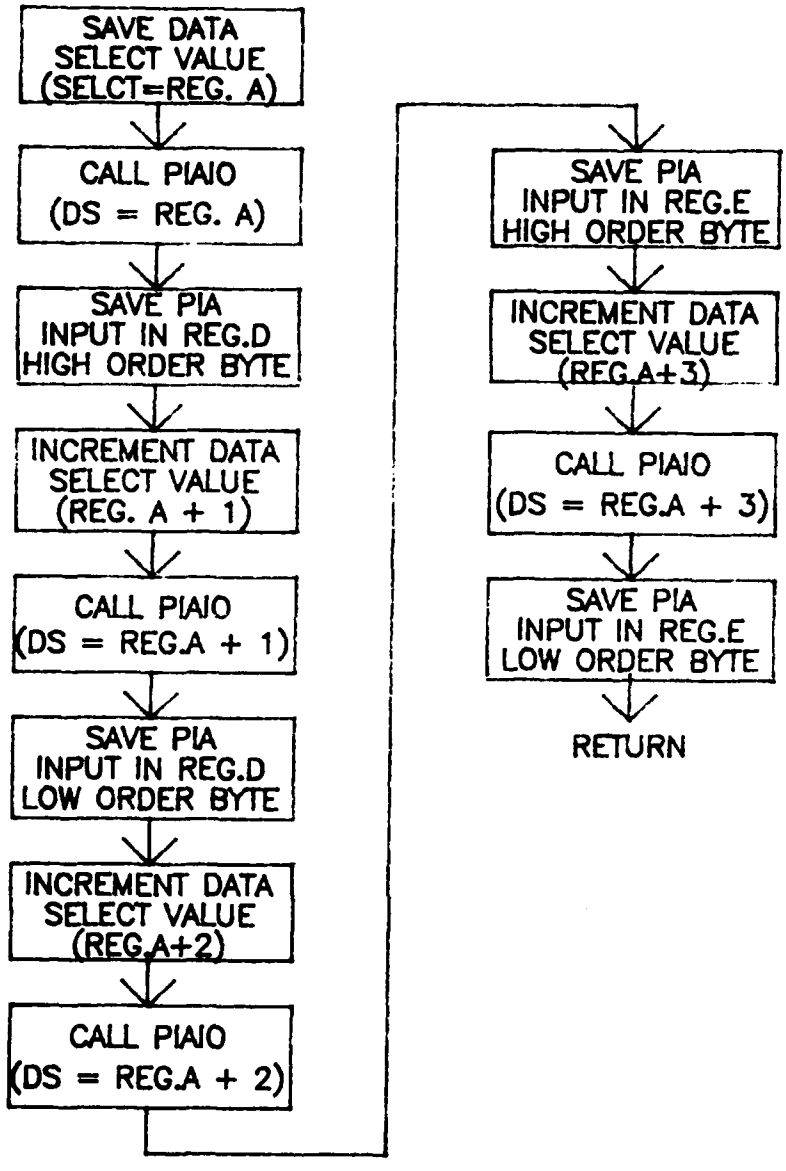
The low level software routines which control this interface are the PIAIO subroutine and the PIAIT interrupt service routine. These routines will now be discussed.

(a) PIAIT INTERRUPT - PIA HANDSHAKE INTERFACE

This is a short routine which interfaces the CA2 handshake strobe to the software. On receiving the CA2 pulse this interrupt is initiated, it resets a flag (FLAGX) which is set by the PIAIO routine before strobing the CA1 contol line. PIAIO sets this flag to "1" before initiating a data transfer with the PIA, it then idles waiting for the CA2 handshake to initiate PIAIT and reset FLAGX at which time the transfer can be terminated. A flow chart for PIAIT is shown in figure 21.

(b) PIAIO SUBROUTINE - PARALLEL INTERFACE CONTROL

This routine in conjunction with PIAIT, provides an interface for the SLUAT and WRTRG routines to the HP parallel port. A flow chart for this routine is shown in figure 22. Its basic function is to carry out the interaction needed with the PIA based on the data select value given in memory location SELCT and on the current mode of the PIA. This routine hides all the handshaking and other PIA dependent functions from



FLOW CHART – RETRV ROUTINE
FIGURE 23.

the higher level routines. This routine is passed data and returns data thru memory location SELCT. The low order 4 bits of SELCT are used to pass the data select value to PIAIO, the high order nibble is used for passing data in or out of the routine. If PIAIO is used in mode 3, to write data to the HP, the data to be output to the HP is put in the high order nibble of SELCT before PIAIO is called. When using PIAIO in this mode the value in SELCT is not destroyed and remains the same upon return. If PIAIO is to be used in modes 0, 1 or 2 or if the data select value is greater than 11 then the data select value need only be passed. In this case the routine will accept input from the PIA and will return that value to the calling routine in the high order nibble of SELCT. When using PIAIO to read data the contents of SELCT are destroyed.

(c) RETRV SUBROUTINE

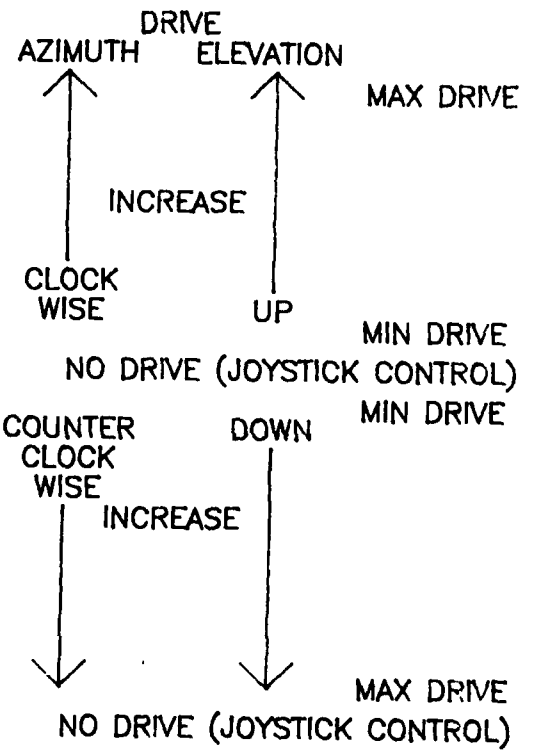
This subroutine uses the PIAIO routine to provide an even higher level of control to the SLUAT routine. This routine is passed a data select value in register A from the SLUAT routine. This data select value and its three predecessors (regA+1, regA+2, regA+3) are written to the PIA. The corresponding data received from the PIA for each of these values are saved in register pair DE for transfer back to SLUAT. This allows the SLUAT routine to easily obtain the entire BCD or binary value of the S-band azimuth, elevation or L-band azimuth using just one call statement. A flow chart for this routine is shown in figure 23.

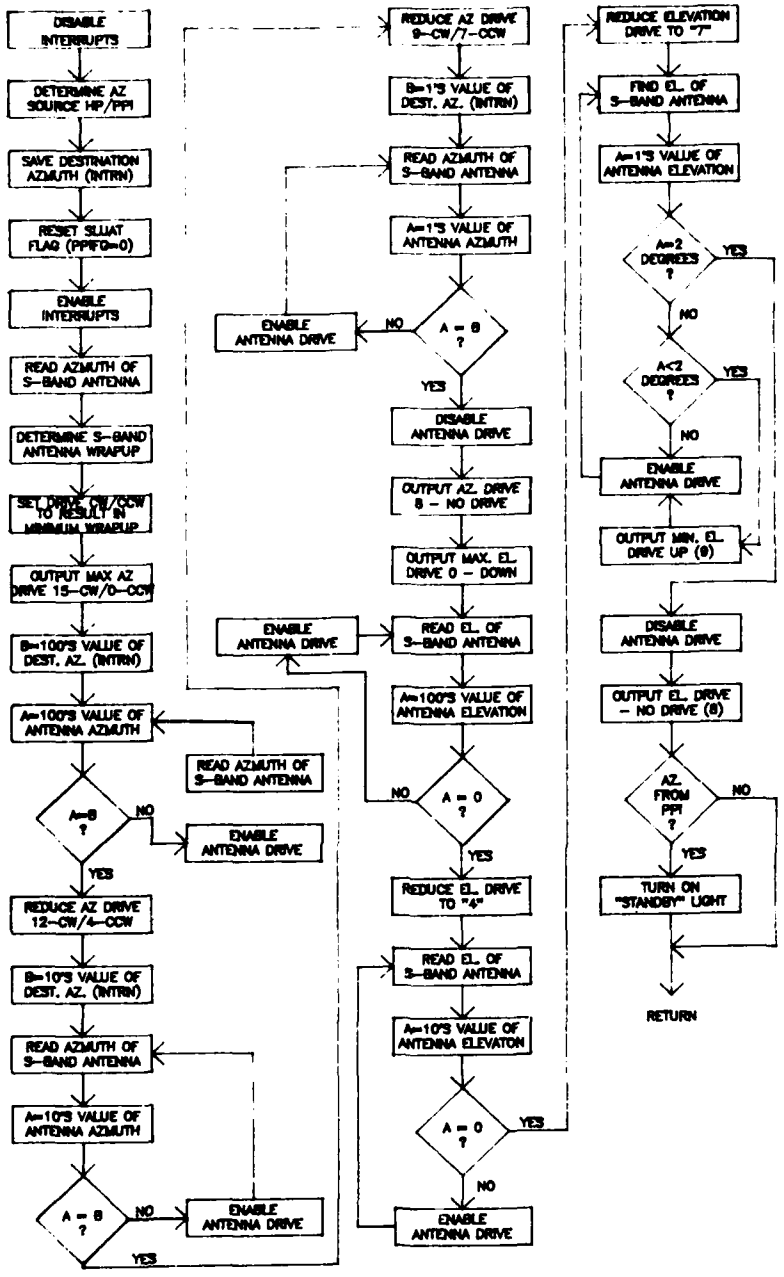
(d) SLUAT ROUTINE - SLEW S-BAND ANTENNA

The purpose of this routine is to drive the S-band antenna to an azimuth position designated in either memory location "BUGAZ" or "GRPAZ" depending on the value of PPIFG. If PPIFG=1 the antenna is driven to the azimuth position contained in "BUGAZ", if PPIFG=2 it is driven to the azimuth position contained in "GRPAZ". The routine drives the antenna by using the D/A-Joystick MUX board which is mounted in the S-band antenna control rack. The board contains two 4-bit D/A converters, one for azimuth and one for elevation the outputs of which are multiplexed with the joystick outputs. These D/A converters are

TABLE 5.
D/A CONVERTERS INPUT/DRIVE

DIGITAL INPUT	ENABLE BIT
15 1 1 1 1	1
14 1 1 1 0	1
13 1 1 0 1	1
12 1 1 0 0	1
11 1 0 1 1	1
10 1 0 1 0	1
9 1 0 0 1	1
8 1 0 0 0	1
7 0 1 1 1	1
6 0 1 1 0	1
5 0 1 0 1	1
4 0 1 0 0	1
3 0 0 1 1	1
2 0 0 1 0	1
1 0 0 0 1	1
0 0 0 0 0	1
X X X X X	0



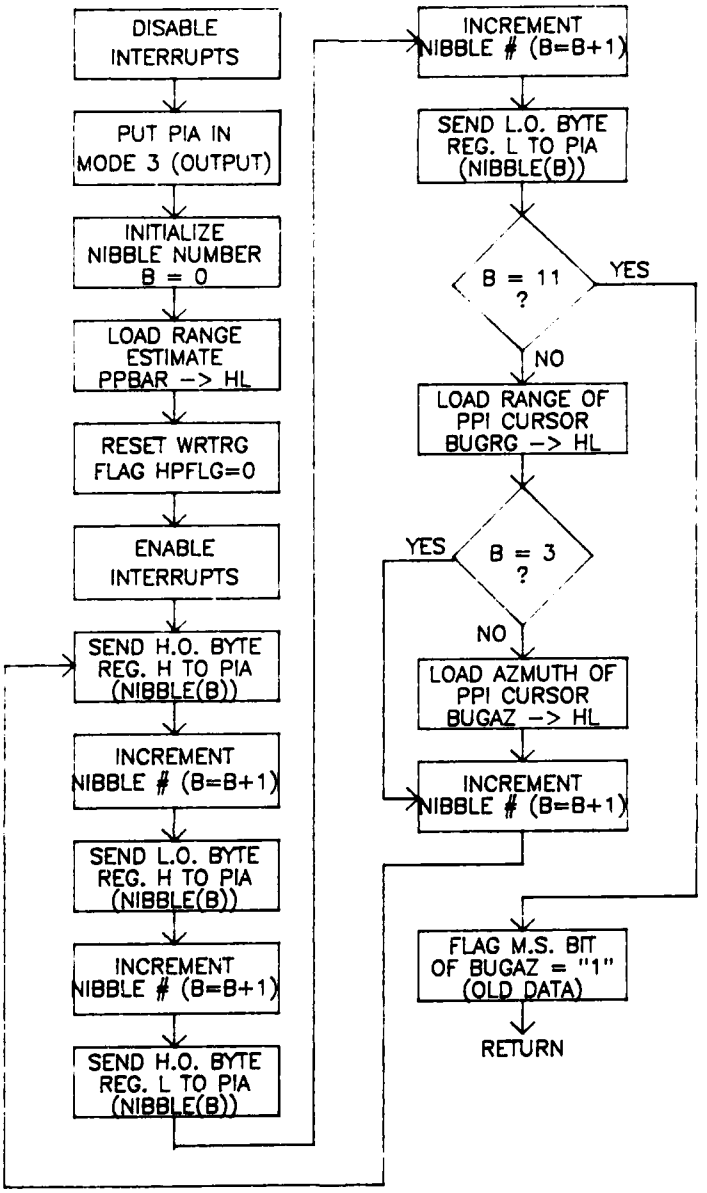


FLOW CHART – SLUAT ROUTINE
FIGURE 24.

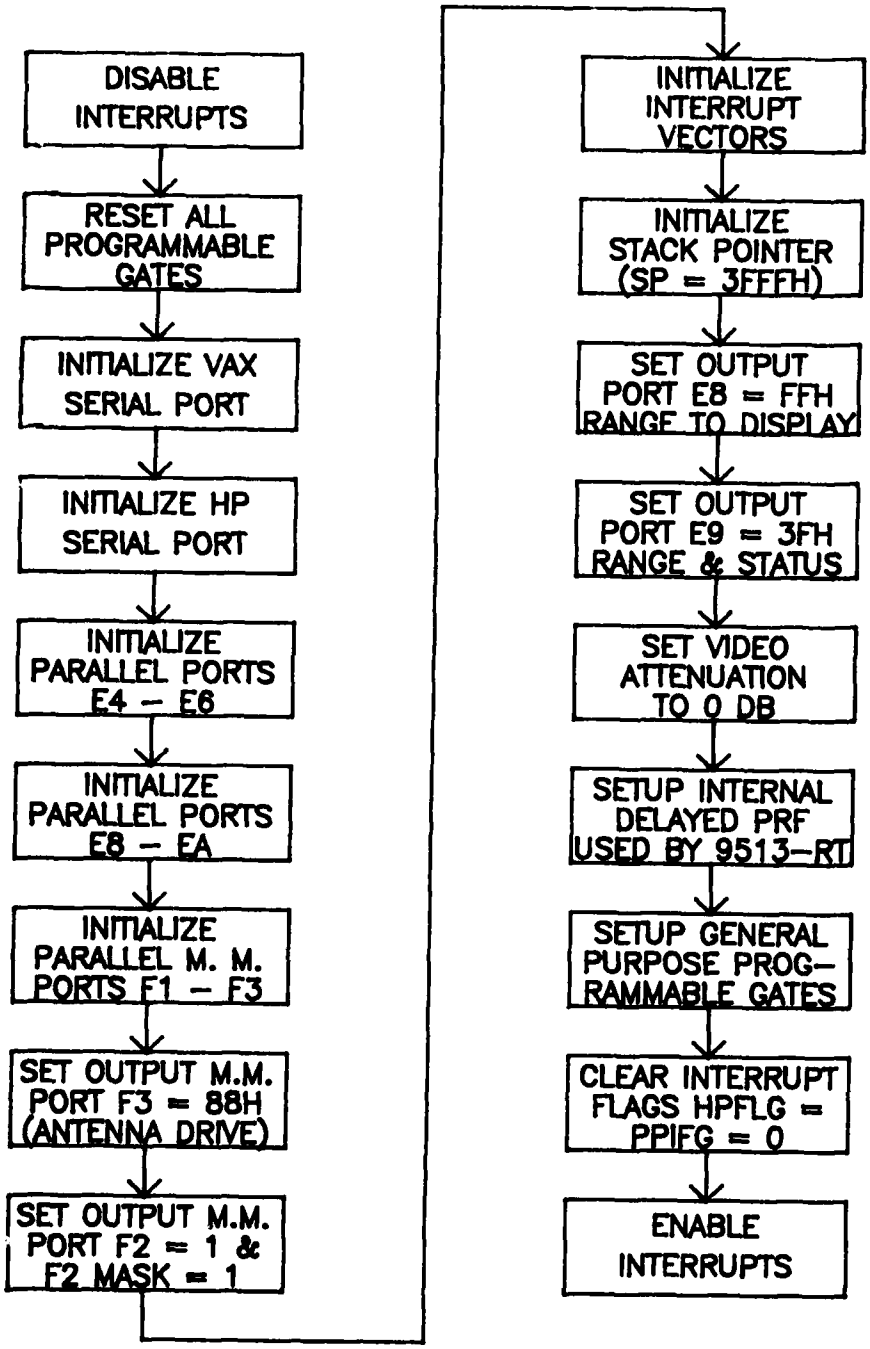
driven by port "F1" hex. The upper four bits of this parallel port are used to drive the elevation D/A converter and the lower four bits to drive the azimuth D/A converter. The multiplexing between the D/A's and the joystick is done in two ways. First, if the input for either of the D/A's is set for zero drive (input=1000b=8) the corresponding output of that D/A is disabled and control is given to the joystick for that channel. Secondly, a control bit is provided to enable/disable the outputs of both D/A's simultaneously. This bit is accessed through port "F2" - bit 2, it is set to "1" to enable the D/A outputs and is reset to disable the D/A's and to give control to the joystick. This bit is changed using a mask associated with port "F2"; bit 2 in this mask is set/reset to obtain the desired result and then the entire mask is written. This is necessary because port "F2" also used for I/O with the HP parallel interface; the values of the bits used for that interface must be preserved.

The input values for each D/A converter and their associated drives are shown in table 5. An input of "15" into the azimuth D/A would give a maximum drive in the clockwise direction, an input of "0" would give a maximum drive in the counter clockwise direction. Similarly, if these inputs were entered into the elevation D/A their respective outputs would be a maximum drive up in elevation ("15") and a maximum drive down in elevation ("0"). It should be noted that if the enable bit is "0" no drive will occur in either azimuth or elevation regardless of the input values. It should also be mentioned that over and above the control bit the azimuth and elevation drives are enabled separately, for example a value of "8" on the azimuth input does not disable the elevation output.

This routine uses both the RETRV and PIAIO routines to communicate with the HP parallel interface. A flow chart for this routine is shown in figure 24.



FLOW CHART – WRTRG ROUTINE
FIGURE 25.



FLOW CHART – INITIALIZATION ROUTINE
FIGURE 26.

(e) WRTRG ROUTINE - WRITE RANGE ESTIMATE TO HP

The purpose of this routine is to write the estimated range of the target and the range and azimuth values of the PPI cursor to the HP computer. The routine sends this information over the parallel interface by using the PIA in mode 3. The format for this transfer is shown in table 4. Each nibble (0 thru 11) is sent in succession starting with the most significant nibble of the binary range estimate. Each nibble is sent in the way described in the first part of this section under mode 3 operation. Contained in the MSN of the PPI cursor azimuth is a flag which should be used by the HP software to determine the status of the PPI cursor information. The most significant bit of this nibble will be set to "0" if new data is present in nibbles 4 thru 11 and it will be set to "1" if the current values have been previously transferred. This bit does not govern the status of the binary range estimate (nibbles 0 - 3), this value is updated just before WRTRG is invoked and is therefore new on every transfer.

This routine uses the PIAIO routine to communicate with the parallel interface and is invoked by either the IRINT or UPDAT routines by setting HPFLG=1. A flow chart for this routine is shown in figure 25.

(5) INITIALIZATION

The initialization routine is responsible for initializing all I/O ports, interrupt vectors, pointers, programmable gates and any presetting or clearing that is required for successful operation of the range tracker and other associated functions. This routine sets up (programs) any general purpose programmable gates and any range tracker supportive gates that are needed at power up. The operation of any of the general purpose programmable gates that are programmed in this routine can be altered by using the "GATES" program to download to the micro (see HP SERIAL LINK - DOWNLOADER and HP SOFTWARE sections). Any future gates that may be needed at power up should be initialized in

this routine in the same manner.

This routine is initiated on power up and by pressing the "RESET" button on the front of the range tracker. A flow chart for this routine is shown in figure 26.

b. HP SOFTWARE

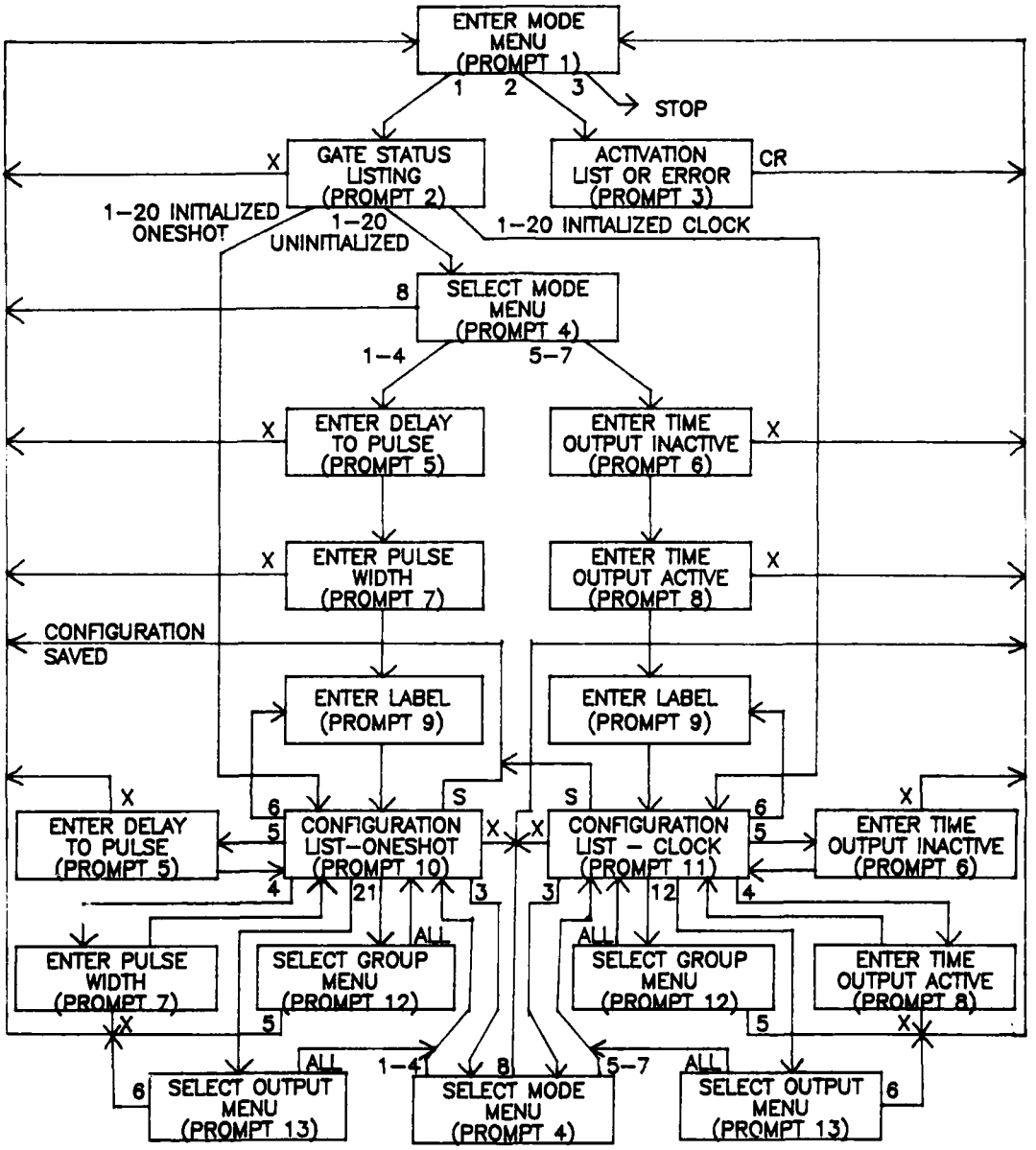
(1) GATES PROGRAM - DOWNLOADER

The HP software for this project consists of one program which provides an interface between the user and the 20 programmable gates. This program consists of a main program and 12 subprograms all written in FORTRAN and uses a version of the HP Terminal Control Library for use on HP and Intertec terminals. The portion of this library that is used in this program is included in the HP software listing in Appendix B.

The program provides a menu-answer type of interaction which sequentially leads the user through all the required steps for programming a gate. The program allows a 16 character label for each of the 20 gates for identification purposes and provides the capability to list all the gates and their corresponding labels at once (unprogrammed gates are labeled "UNINITIALIZED"). The program also provides a means to change the labels or configuration of any of the gates by just a few keystrokes on the terminal.

Figure 27 shows the flow from one menu to the next corresponding to the various inputs. Each arrow leading from a given "menu box" is labeled with the input which corresponds to that menu to menu transition. Any inputs that are not shown on this graph are ignored by the menu in question (ie: any inputs other than 1, 2 or 3 for the ENTER MODE menu (prompt 1) will cause this menu to be displayed again). An example of each of the 13 different prompts are listed at the end of Appendix B.

When using this program to set up the programmable gates two things should be kept in mind: 1. A gate configuration cannot be downloaded unless it is first saved; 2. After making a change to a particular gate and saving it, the results will not be seen at the gate outputs until a download is done. By remembering these two things and by using the flow chart (figure 27) and the prompts in Appendix B the use of this program should come very easy.



MENU FLOW CHART
FIGURE 27.

5. CONCLUSION

The range tracker and programmable gates were designed, built and installed during the period from December 1981 to November 1982. The tracker has been successfully demonstrated several times tracking live radar target returns and has proved to be completely compatible with the S-band angle tracker. The programmable gates have proved to be very useful for providing timing control for the angle tracker and also for testing and debugging various pieces of hardware which require the use of a pulse generator. The ease in configuring these gates has made them very useful for this purpose.

The hardware and software for this project are well documented in this report and also in the software listings themselves. This documentation should prove to be very useful in understanding the the internal workings of the range tracker and also in debugging or modifying the hardware and software should the need arise.

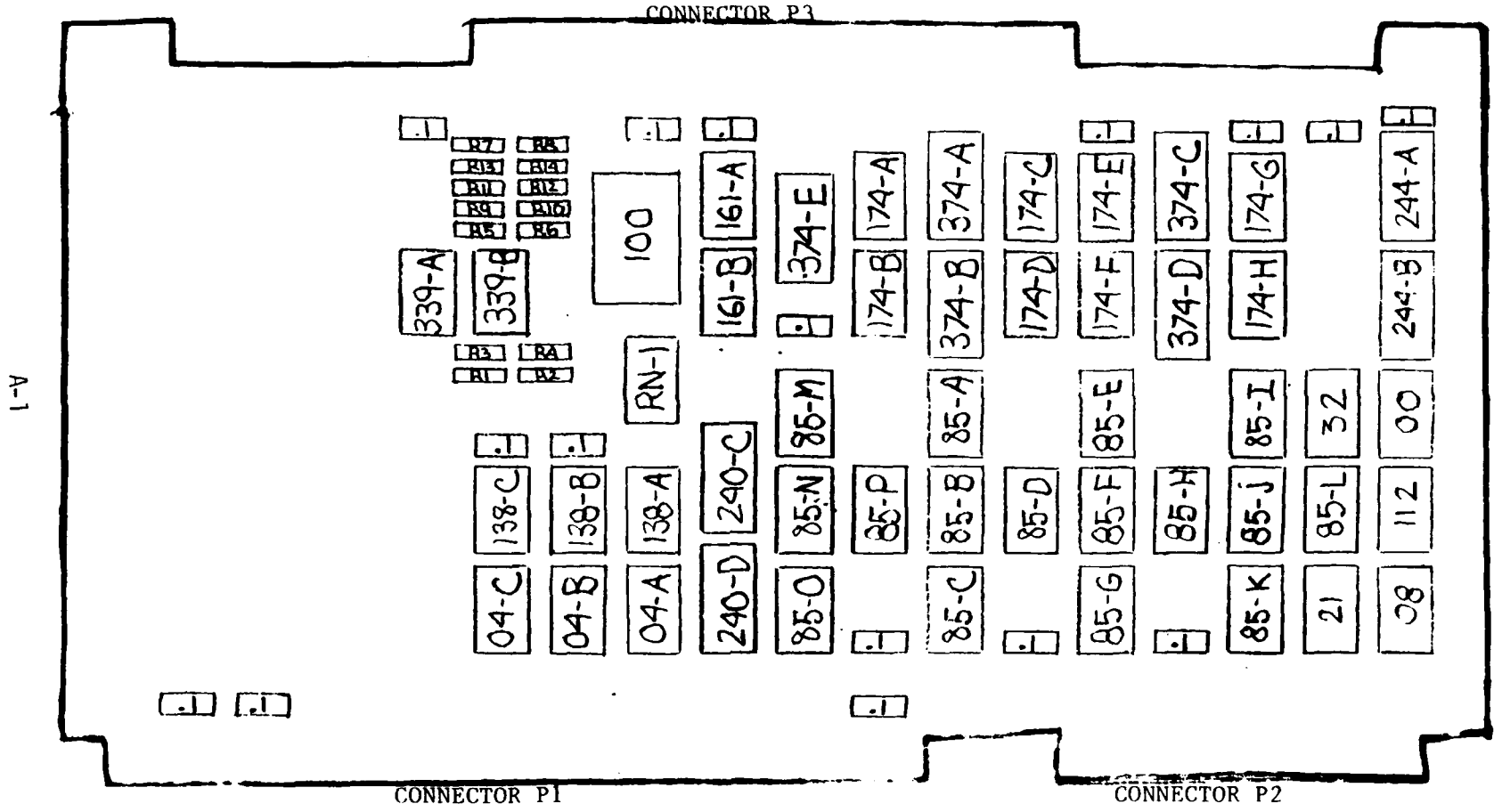
APPENDIX A

HARDWARE SCHEMATICS

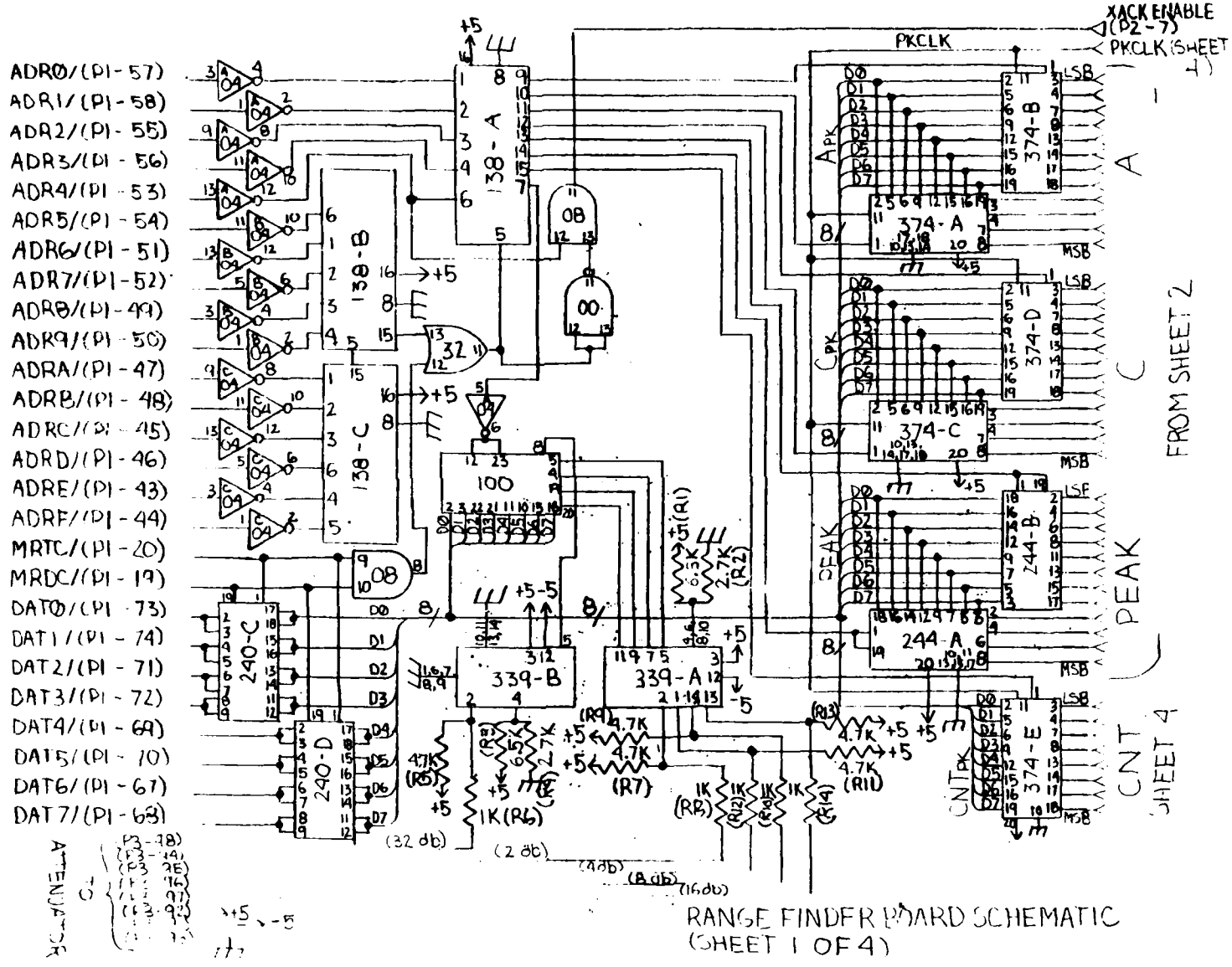
TIMING DIAGRAMS

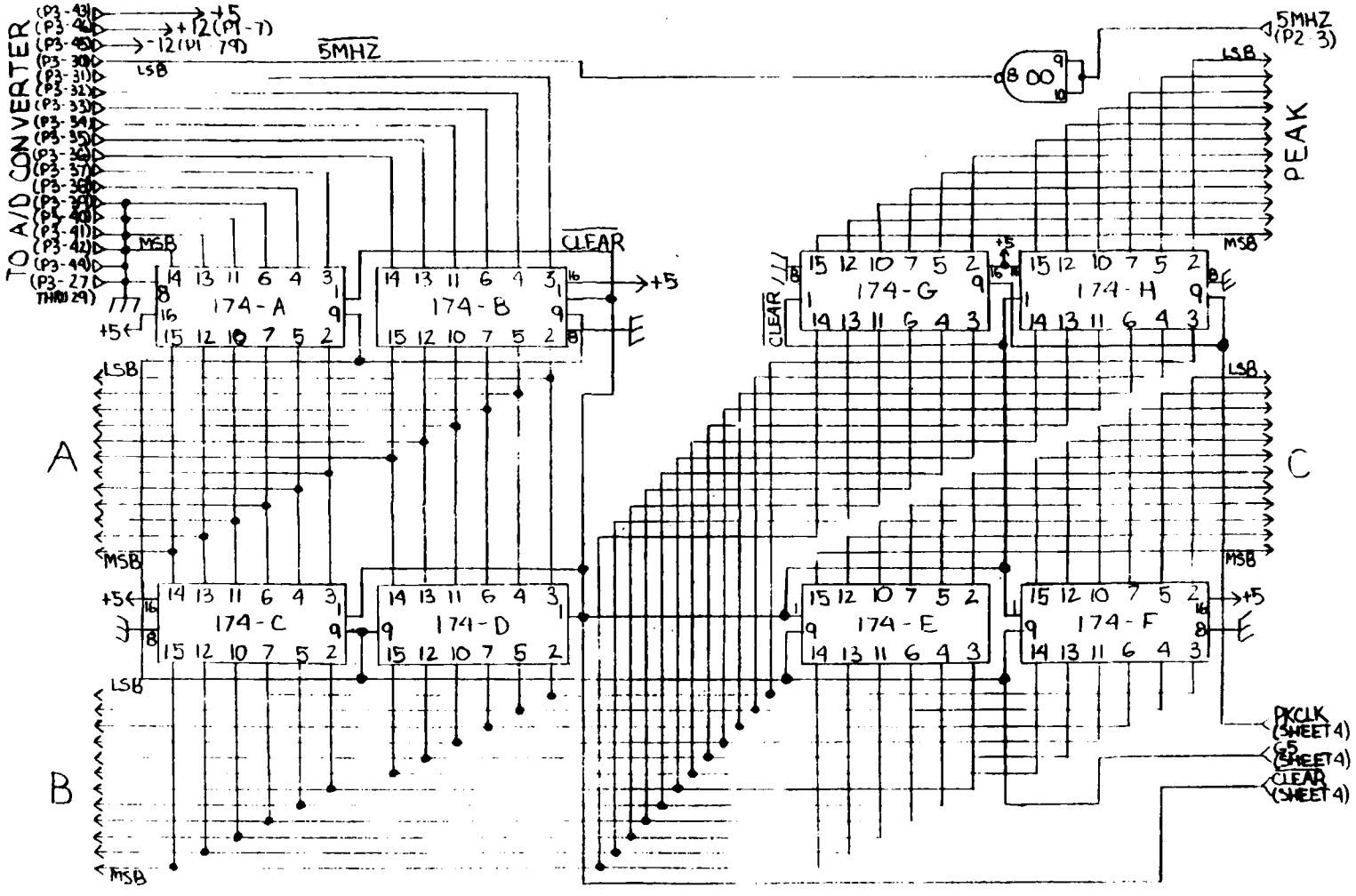
&

CABLING DIAGRAMS

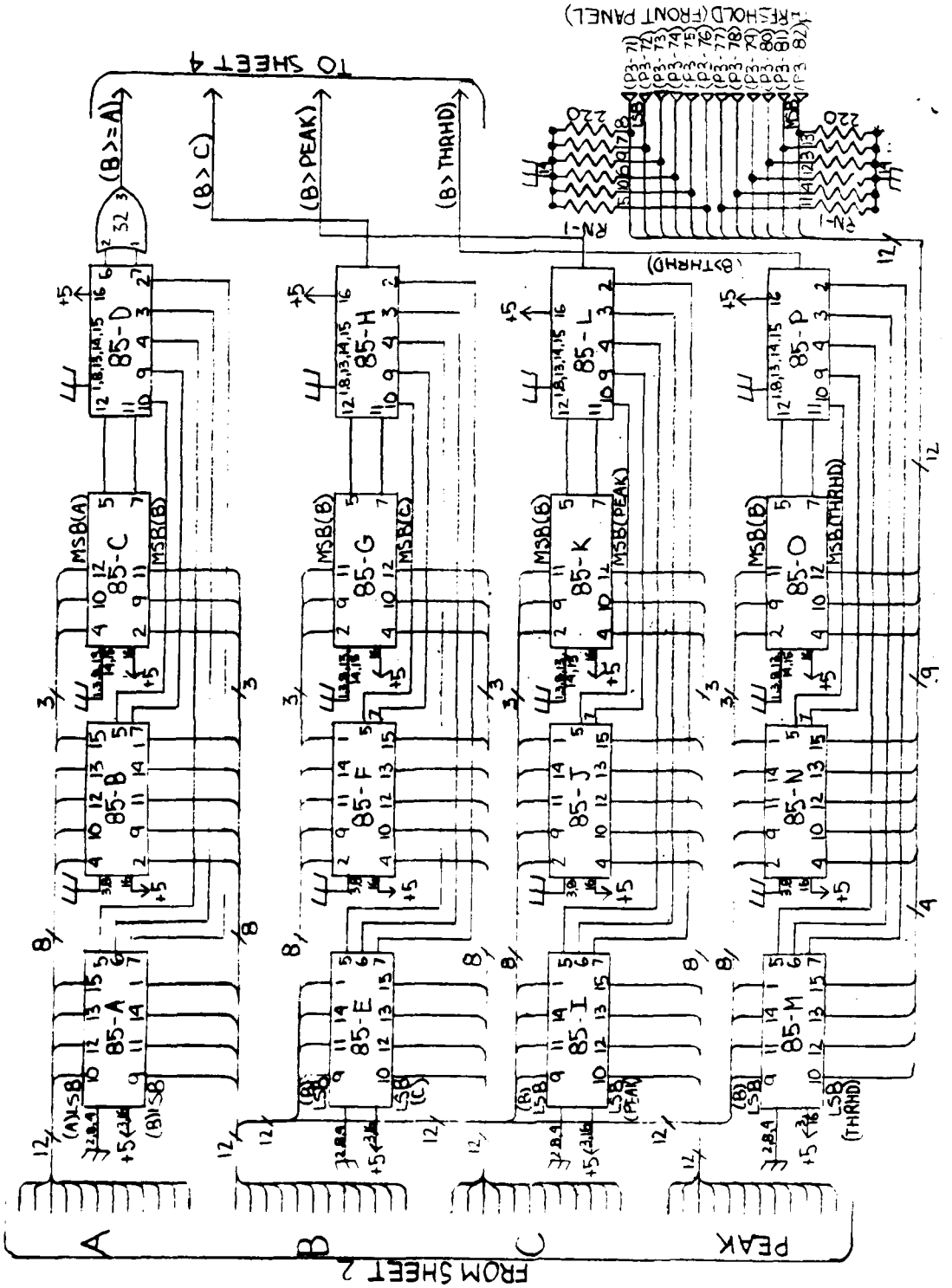


RANGE FINDER BOARD LAYOUT

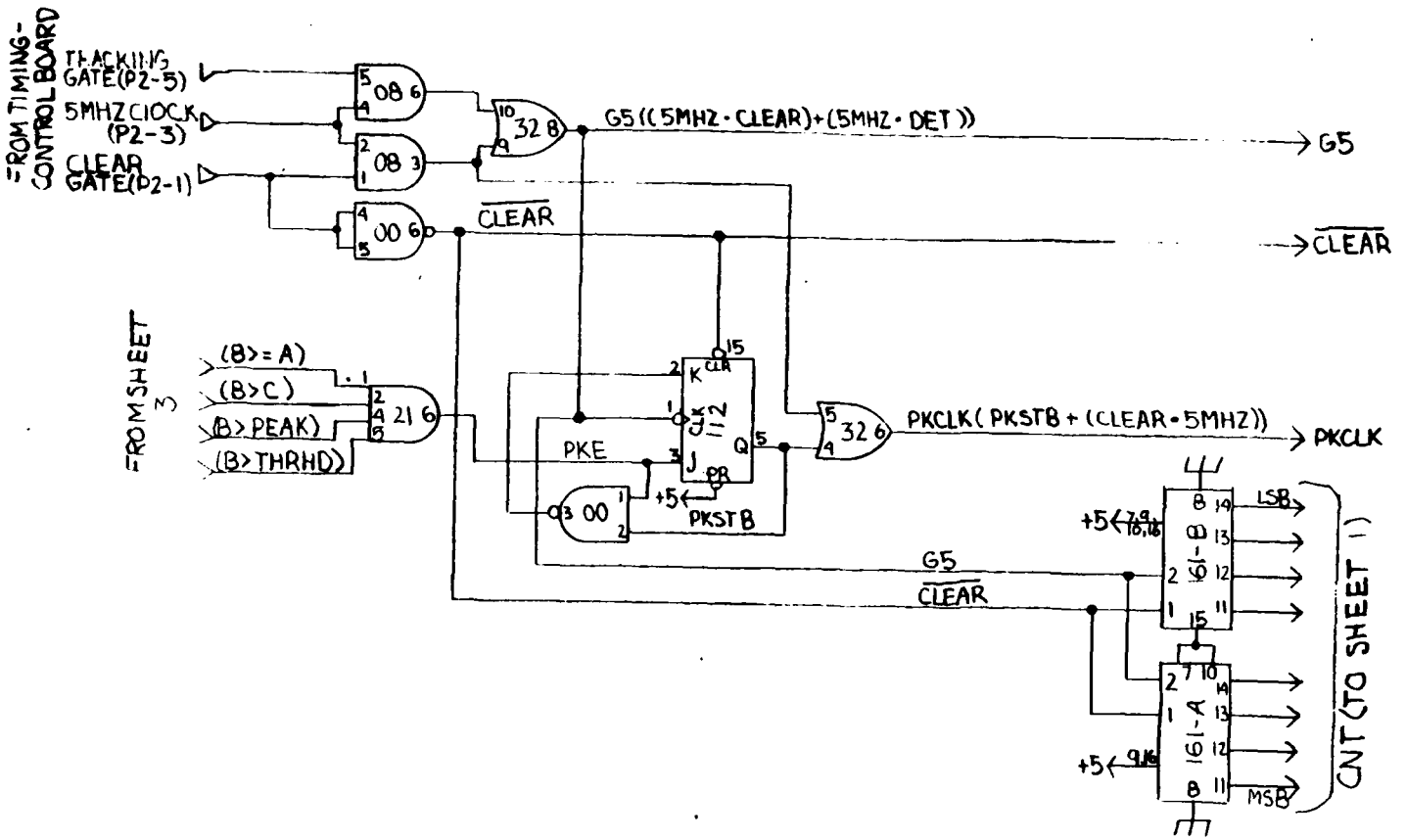

ADR0/(P1-57)
ADR1/(P1-58)
ADR2/(P1-55)
ADR3/(P1-56)
ADR4/(P1-53)
ADR5/(P1-54)
ADR6/(P1-51)
ADR7/(P1-52)
ADR8/(P1-49)
ADR9/(P1-50)
ADRA/(P1-47)
ADRB/(P1-48)
ADRC/(P1-45)
ADRD/(P1-46)
ADRE/(P1-43)
ADRF/(P1-44)
MRTC/(P1-20)
MRDC/(P1-19)
DAT0/(P1-73)
DAT1/(P1-74)
DAT2/(P1-71)
DAT3/(P1-72)
DAT4/(P1-69)
DAT5/(P1-70)
DAT6/(P1-67)
DAT7/(P1-68)
XACK ENABLE (P2-7)
PKCLK
PKCLK (SHEET 4)
FROM SHEET 2
PEAK
CNT
SHEET 4
RANGE FINDER BOARD SCHEMATIC
(SHEET 1 OF 4)



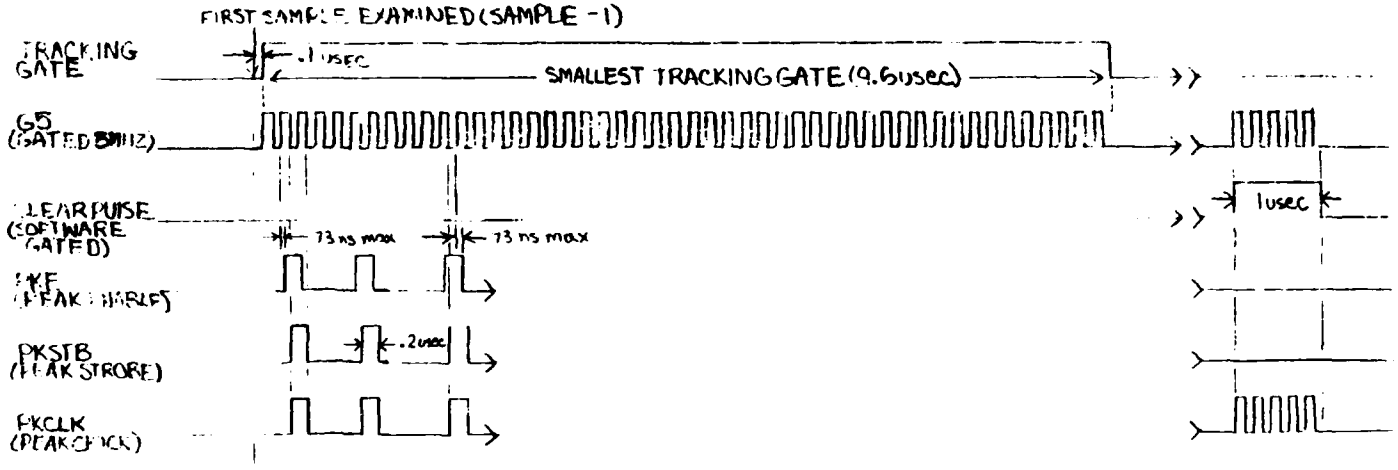
RANGE FINDER BOARD SCHEMATIC
(SHEET 2 OF 4)



RANGE FINDER BOARD SCHEMATIC
(SHEET 3 OF 4)

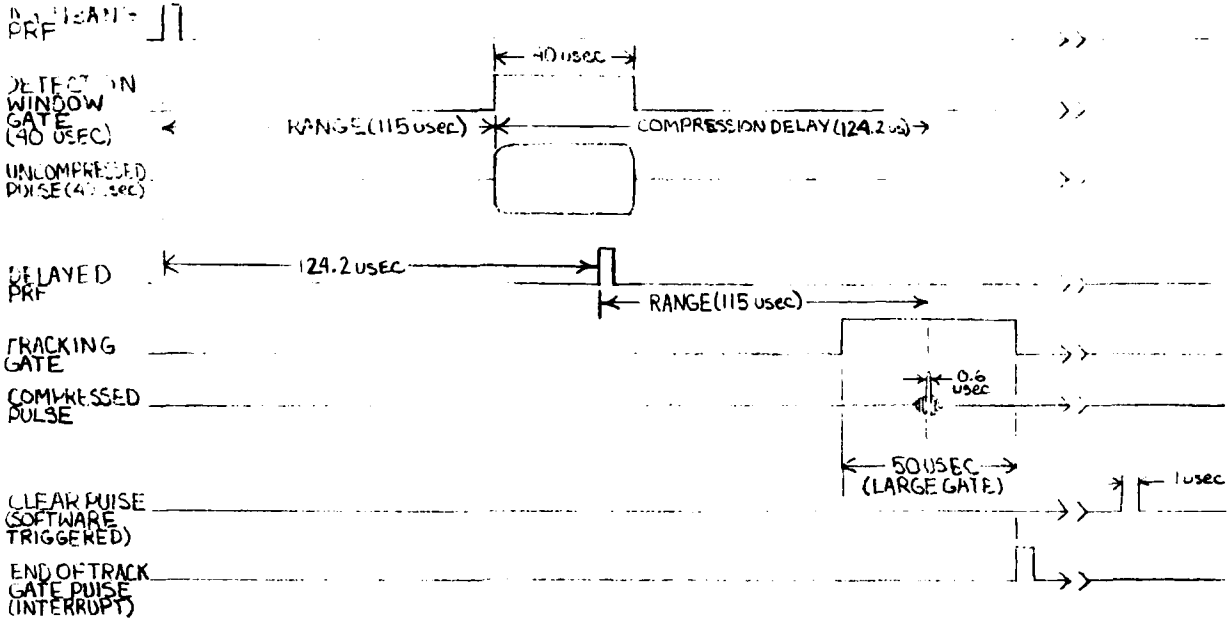


RANGE FINDER BOARD SCHEMATIC
(SHEET 4 OF 4)



A/D (SAMPLE #)	-3 -2 -1 1 2 3 4 5 6 7 8 9 10 11	- A/D SAMPLES TAKEN ON FALLING EDGE OF 5MHZ CLOCK.
A (SAMPLE #)	Ø -1 1 2 3 4 5 6 7 8 9 10	ASSUMING THRESHOLD IS ZERO. TO HAVE A PEAK: (A<=B>C) AND (B>PEAK)
B (SAMPLE #)	Ø Ø -1 1 2 3 4 5 6 7 8 9	
C (SAMPLE #)	Ø Ø Ø -1 1 2 3 4 5 6 7 8	
A_{PK} (SAMPLE #)	Ø Ø 1 1 1 1 5 5 5 5 10	THE VALUES OF THESE THREE SAMPLES ARE READ IN BY THE COMPUTER AND ARE FITTED WITH A PARABOLA TO DETERMINE THE TRUE POSITION OF THE PEAK TO 0.1 usec (1/6 range cell).
PEAK (SAMPLE #)	Ø Ø 1 1 1 1 4 4 4 4 4 9	
C_{PK} (SAMPLE #)	Ø Ø Ø Ø Ø Ø 3 3 3 3 8	
CNT	Ø 1 2 3 4 5 6 7 8 9 10 11	
CNT_{PK}	Ø Ø 2 2 2 2 6 6 6 6 6 11	- AFTER INTERROGATION BY THE COMPUTER THE POSITION OF THE PEAK (CNTPK) IS DECREMENTED BY 2. IF THE RESULT IS ZERO THE PEAK IS IGNORED FOR REASONS SHOWN HERE. (THE PRESET VALUE Ø ENTERS INTO THE COMPARISON)
PEAK DETECTED	*	*		*
REASON FOR PEAK OR NO PEAK	S(-1) ≥ Ø	S(-1) > S(1)	S(4) > S(3)	S(1) > S(4)

RANGE FINDER TIMING

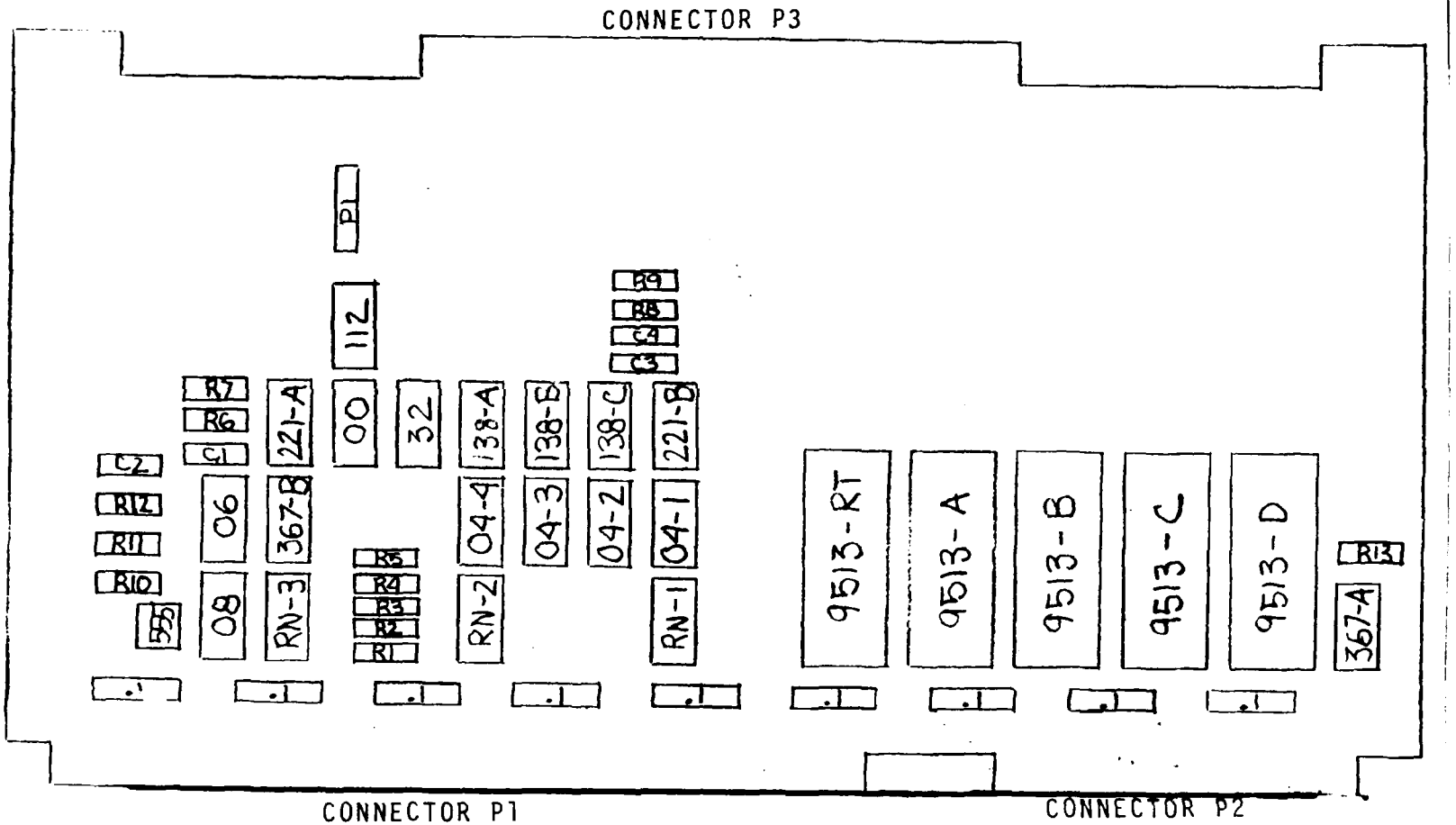


NOTE: THE 40 USEC DETECTION WINDOW GATE IS TO BE USED BY THE ANGLE TRACKER. IT REMAINS 40 USEC WIDE REGARDLESS OF THE TRACKING ERROR AND IS CENTERED AROUND AND MOVES WITH THE TARGET. THE TRACKING GATE IS USED BY THE RANGE TRACKER TO TRACK THE TARGET. IT ALSO IS CENTERED AROUND AND MOVES WITH THE TARGET BUT THE WIDTH OF THE GATE VARIES WITH THE TRACKING ERROR.

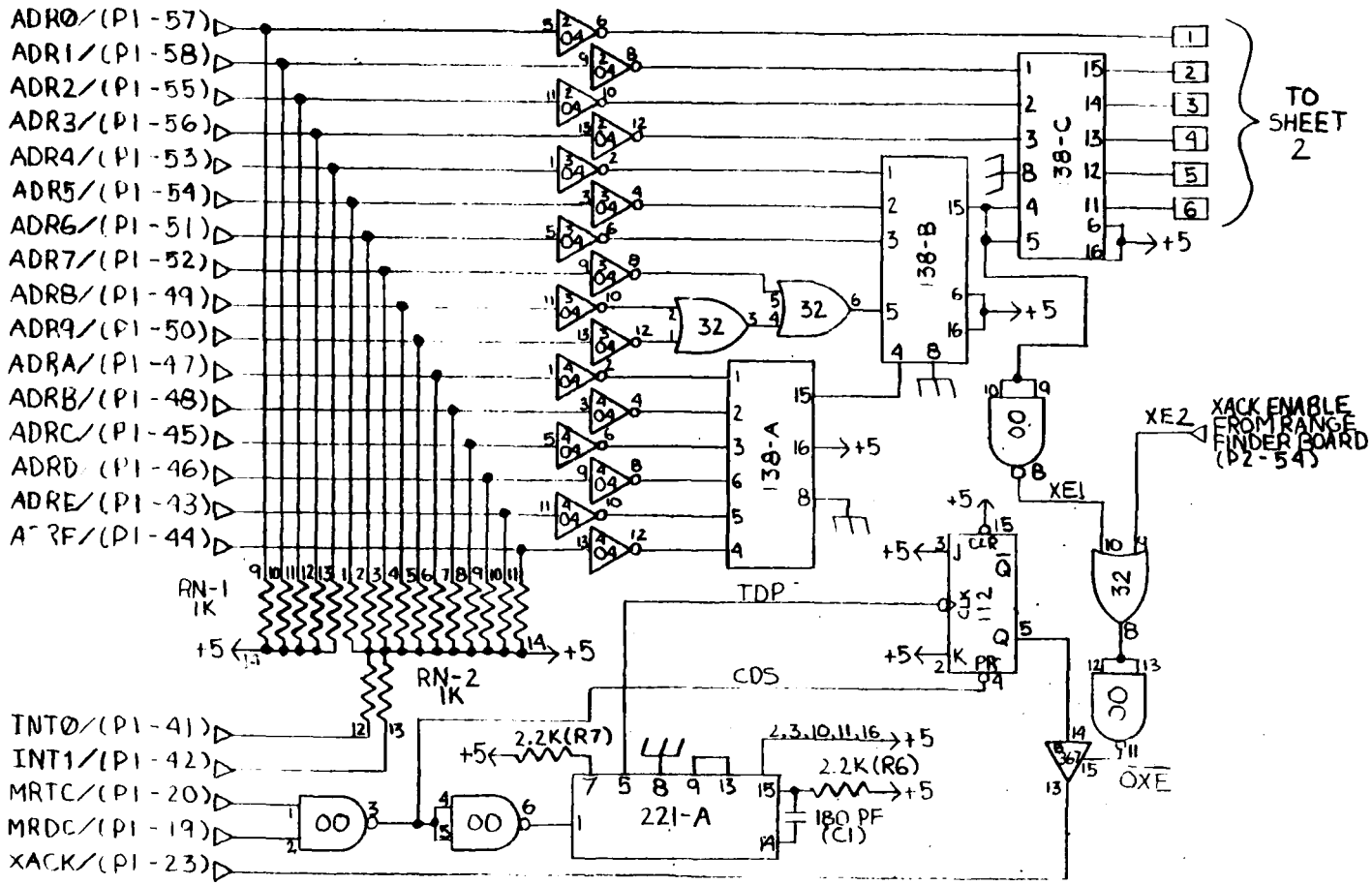
BASIC TIMING FOR SYSTEM INTERFACE

CABLING FOR RANGE--FINDER BOARD

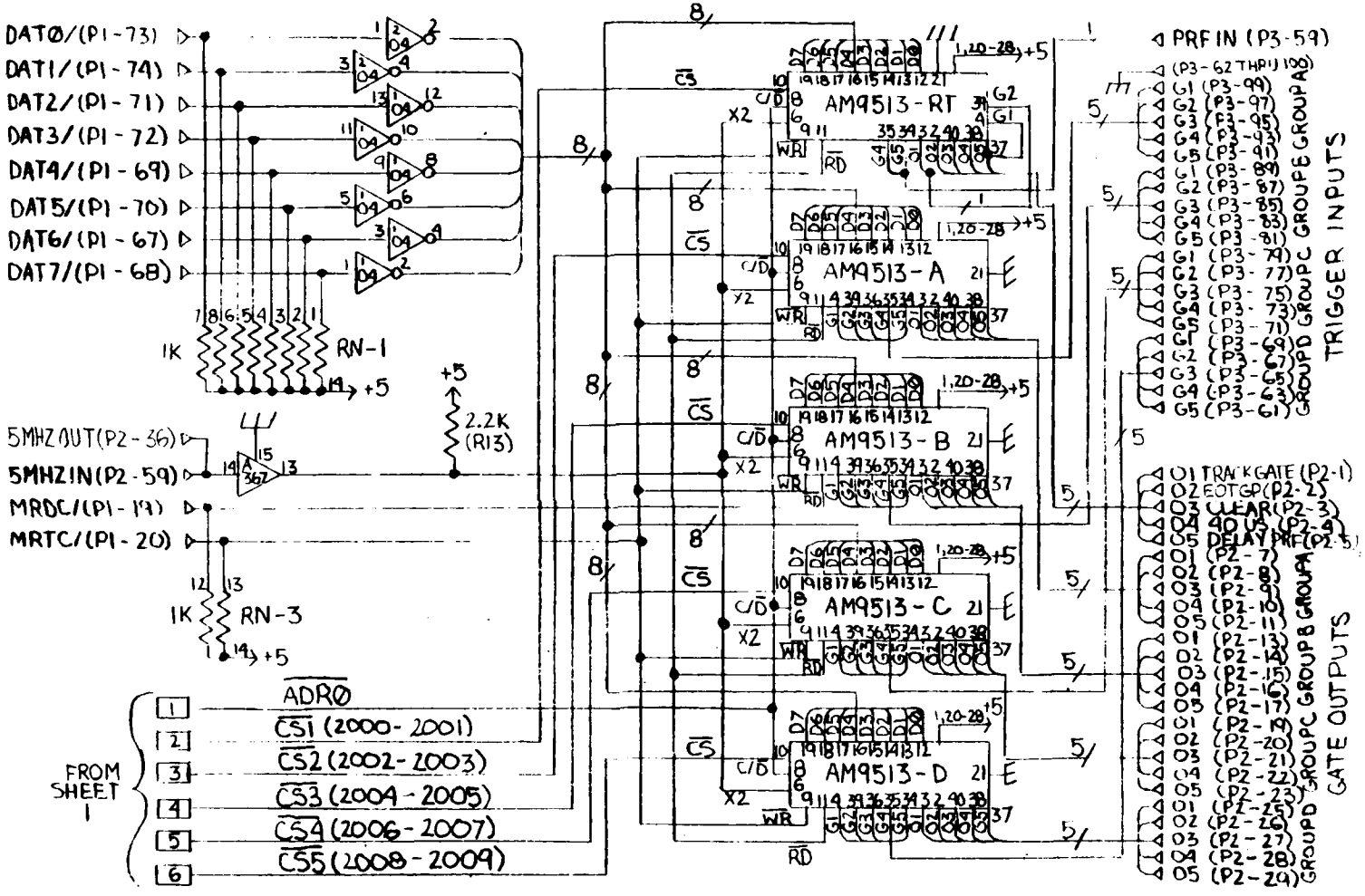
CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
P1	1–86	MULTIBUS STANDARD	TO TIMING–CONTROL BOARD & uCOMPUTER
P2	1	CLEAR GATE	TIMING–CONTROL BOARD (P2 – 3)
	3	5MHZ IN	TIMING–CONTROL BOARD (P2 – 36)
	5	TRACKING GATE	TIMING–CONTROL BOARD (P2 – 1)
	7	XACK HANDSHAKE ENABLE	TIMING–CONTROL BOARD (P2 – 54)
P3	71–82	THRESHOLD INPUT (PIN 71–LSB, 82–MSB)	HEX THUMBWHEEL SWITCH (FRONT PANEL)
	43	+5	A/D CONVERTER (PINS M, W & R)
	45	–12	A/D CONVERTER (PIN E)
	46	+12	A/D CONVERTER (PIN H)
	30	5 MHZ	A/D CONVERTER (PIN N)
	31–42	A/D INPUT (12 BITS) (PIN 31–LSB, 42–MSB) (PINS 39–42 N/C – GND)	A/D CONVERTER (PINS P,Z,Y,X,V,U,T,S RESPECTFULLY)
	4,27–29	GROUND	A/D CONVERTER (DGND) (PINS 11–22)



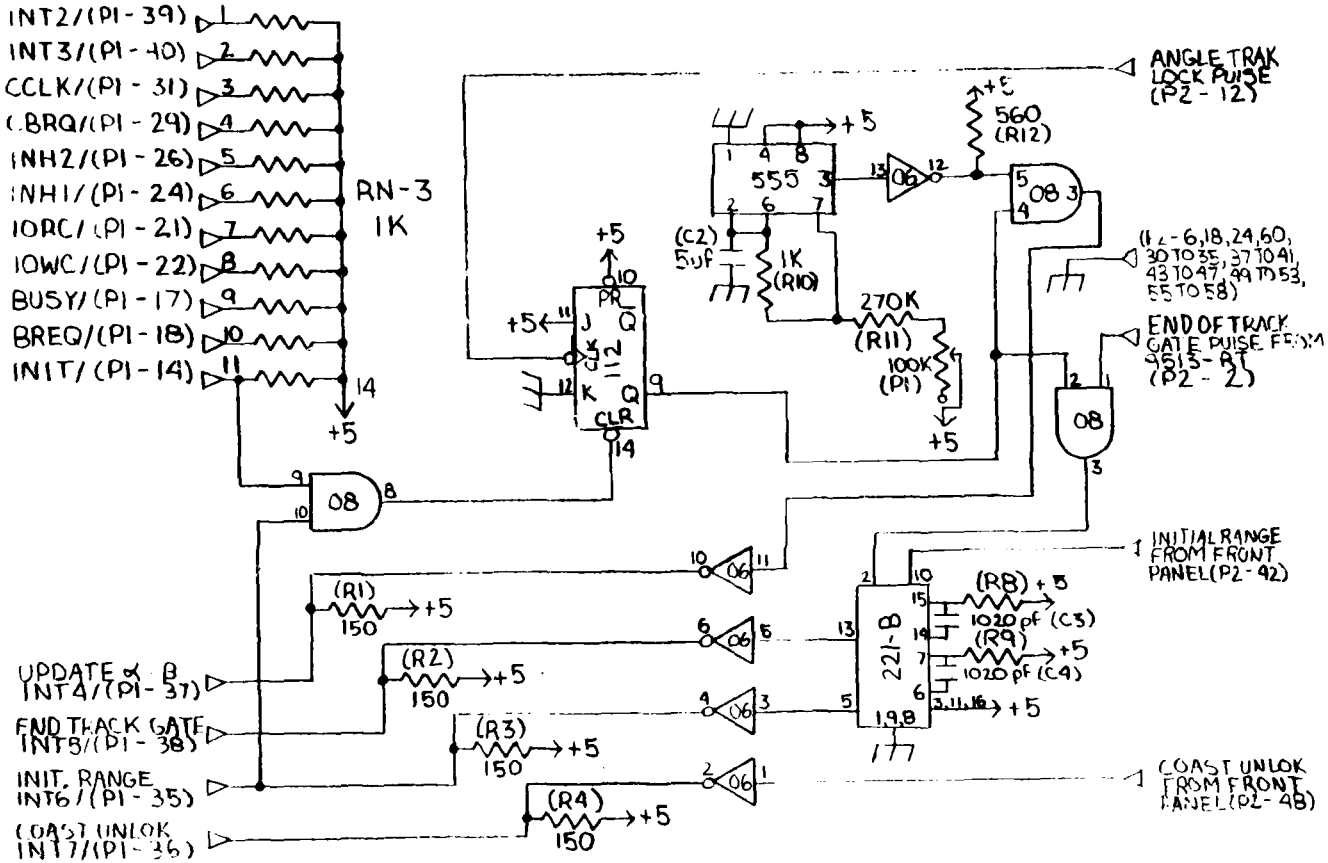
TIMING-CONTROL BOARD LAYOUT



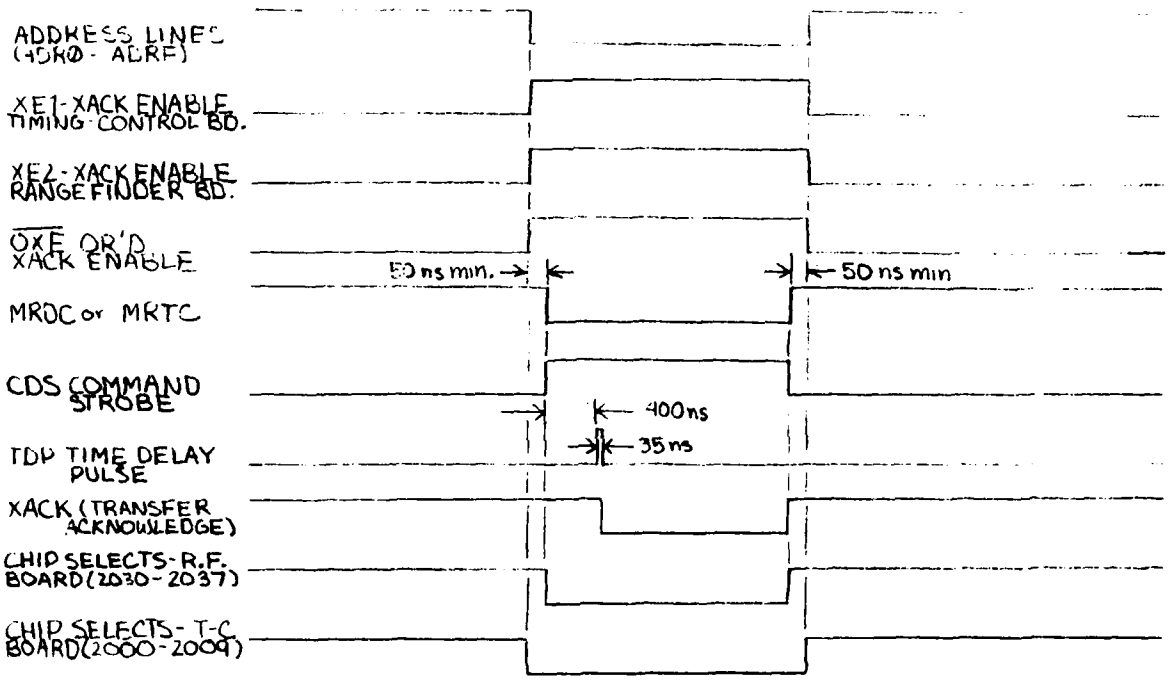
TIMEING CONTROL BOARD SCHEMATIC
(SHEET 1 OF 3)



TIMEING-CONTROL BOARD SCHEMATIC
(SHEET 2 OF 3)



TIMEING-CONTROL BOARD SCHEMATIC
(SHEET 3 OF 3)



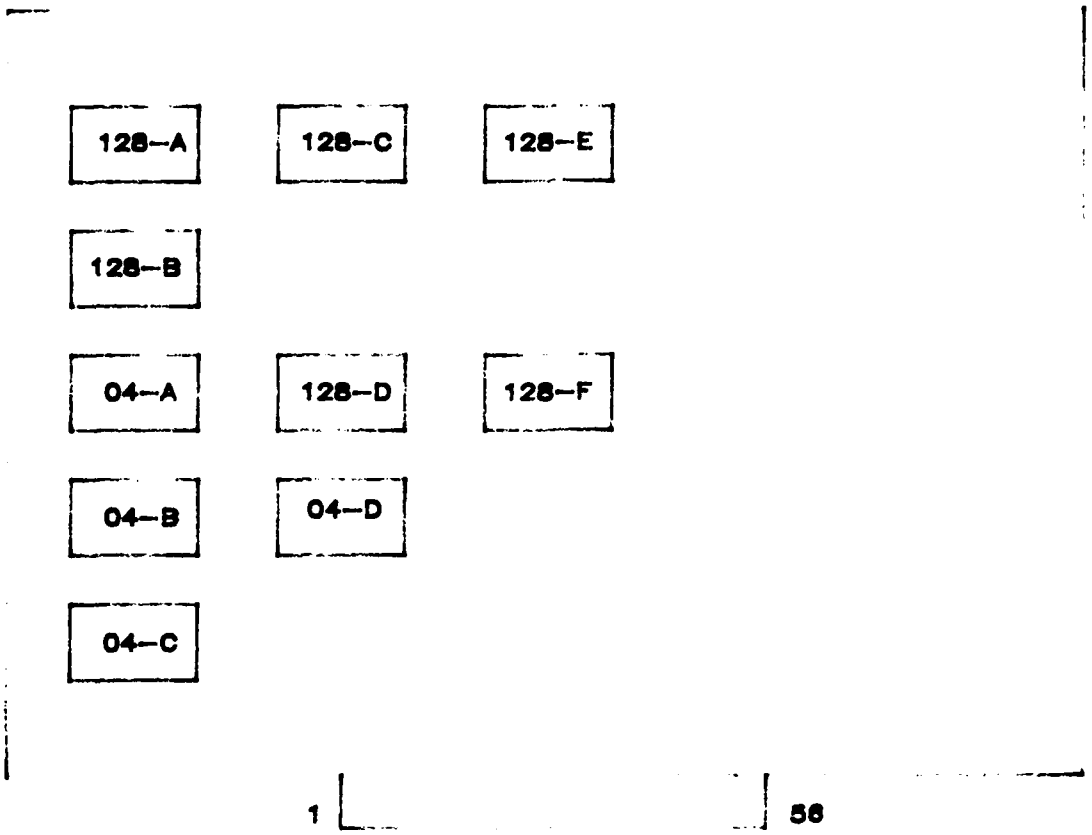
ADDRESS DECODING AND HANDSHAKING

CABLING FOR TIMING–CONTROL BOARD

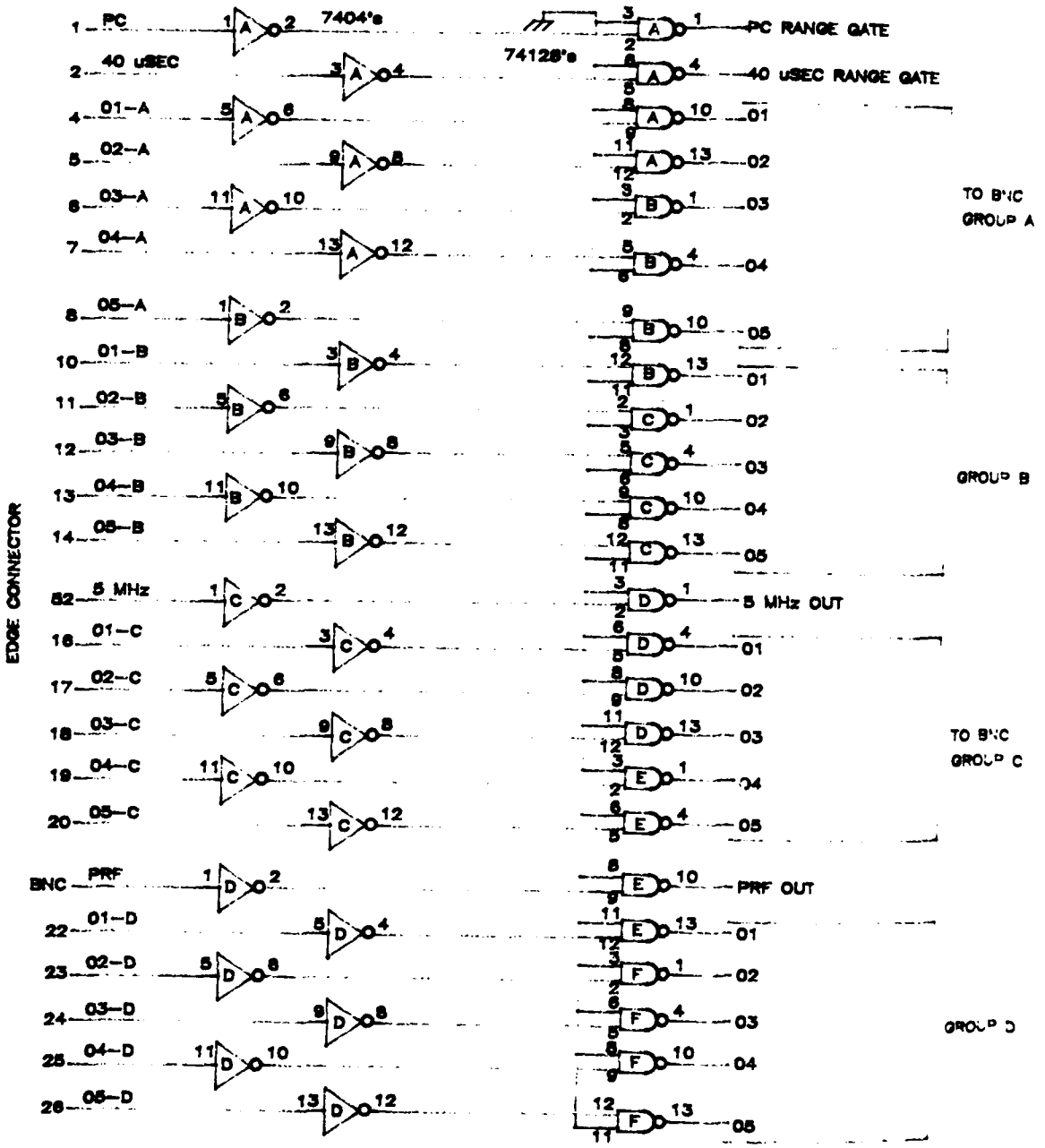
CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
P1	1–86	MULTIBUS STANDARD	TO RANGE–FINDER BOARD & uCOMPUTER
P2	1	TRACKING GATE	RANGE–FINDER BOARD (P2 – 5) & DRIVER BOARD (PIN 1)
	2	END OF TRACKING GATE PULSE	N/C
	3	CLEAR GATE	RANGE–FINDER BOARD (P2 – 1)
	4	40 USEC RANGE GATE	DRIVER BOARD (PIN 2)
	5	DELAYED PRF	N/C
	7–11	O1 – O5 OUTPUTS FROM 9513 – A	DRIVER BOARD (PINS 4 – 8)
	12	ANGLE TRACK LOCK STROBE IN	DRIVER BOARD (PIN 9)
	13–17	O1 – O5 OUTPUTS FROM 9513 – B	DRIVER BOARD (PINS 10 – 14)
	19–23	O1 – O5 OUTPUTS FROM 9513 – C	DRIVER BOARD (PINS 16 – 20)
	25–29	O1 – O5 OUTPUTS FROM 9513 – D	DRIVER BOARD (PINS 22 – 26)
	36	5 MHZ OUT	RANGE–FINDER BOARD (P2 – 3)
	42	INITIAL RANGE INTERRUPT STROBE	PUSH BUTTON FRONT PANEL
	43	COAST UNLOCK INTERRUPT STROBE	PUSH BUTTON FRONT PANEL
	54	XACK HANDSHAKE ENABLE IN	RANGE–FINDER BOARD (P2 – 7)
	59	5 MHZ IN	DRIVER BOARD (PIN 52)
	6,18,24, 30–35, 37–41, 43–47, 49–53, 55–58,60	GROUND	DRIVER BOARD (PINS 3,15,21,27–51)

CABLING FOR TIMING–CONTROL BOARD

CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
P3	91–99 (ODD)	G1–G5 TRIGGER INPUTS 9513–A	PATCH PANEL CONNECTOR PINS (11–19 ODD)
	81–89 (ODD)	G1–G5 TRIGGER INPUTS 9513–B	PATCH PANEL CONNECTOR PINS (21–29 ODD)
	71–79 (ODD)	G1–G5 TRIGGER INPUTS 9513–C	PATCH PANEL CONNECTOR PINS (31–39 ODD)
	61–69 (ODD)	G1–G5 TRIGGER INPUTS 9513–D	PATCH PANEL CONNECTOR PINS (41–49 ODD)
	59	PRF INPUT	PATCH PANEL CONNECTOR PIN 51
	62–100 (EVEN)	GND	PATCH PANEL CONNECTOR PINS (12–52 EVEN)



DRIVER BOARD LAYOUT

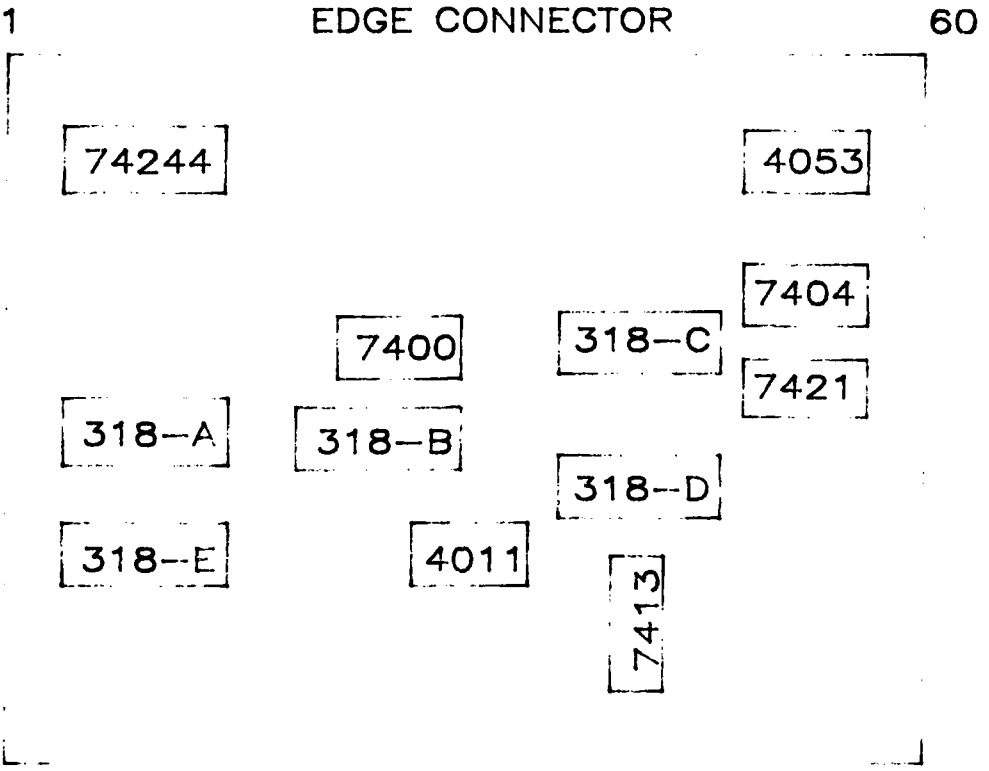


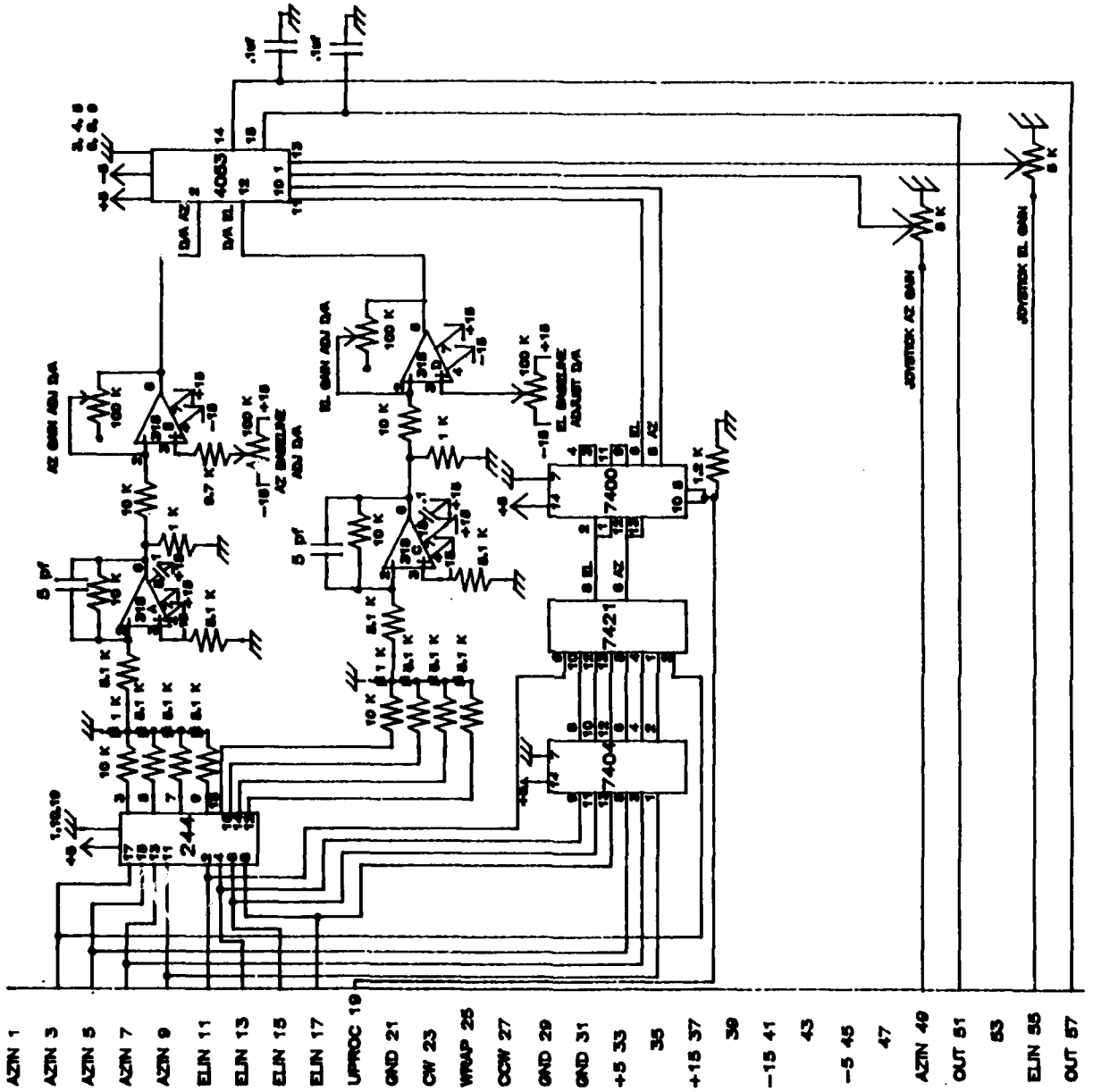
DRIVER BOARD SCHEMATIC

CABLING FOR DRIVER BOARD

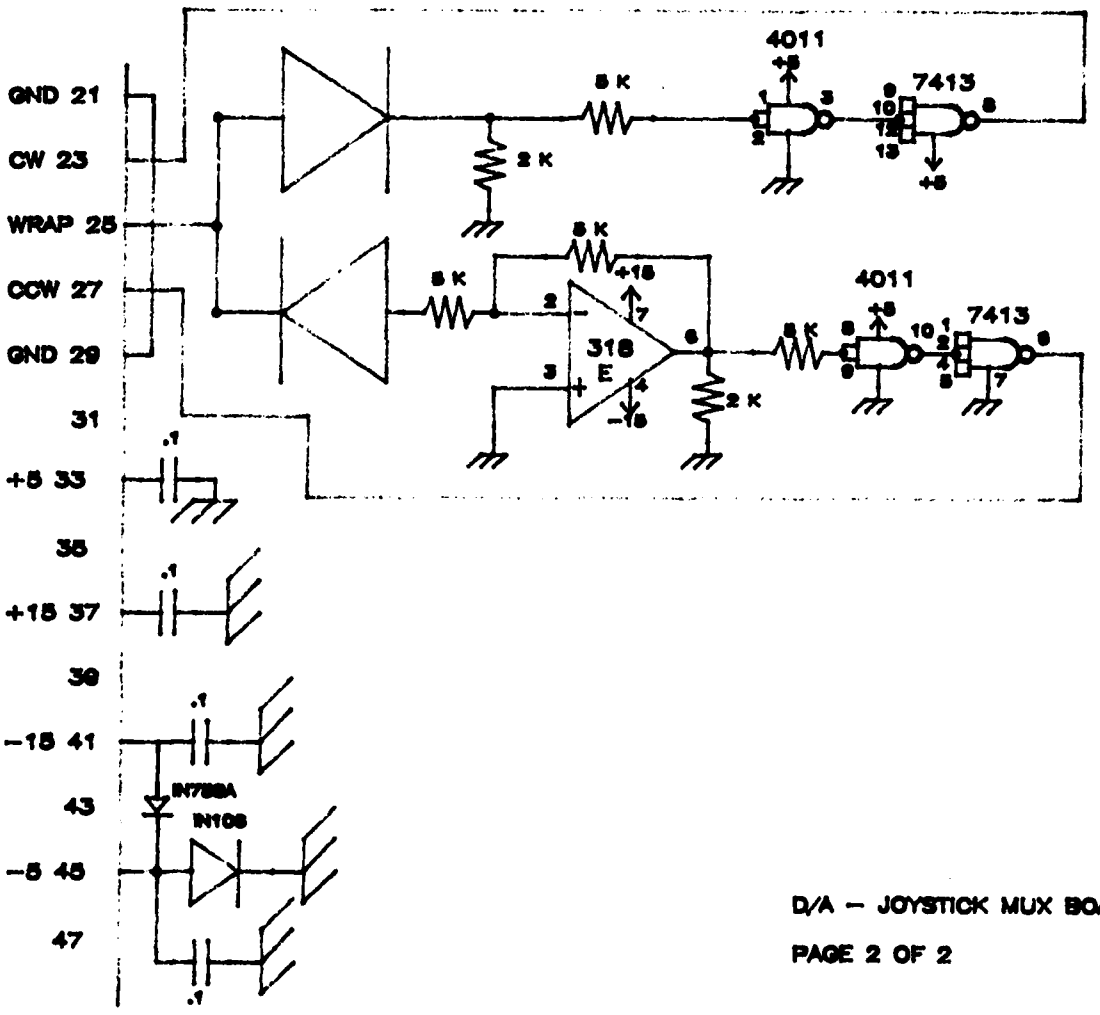
CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
DRIVER BOARD	1–52	SEE CABLING FOR TIMING–CONTROL BOARD	
	54	+5 VOLTS	MULTIBUS PIN 83
	53	GROUND	MULTIBUS PIN 85

D/A JOYSTICK MUX BOARD LAYOUT



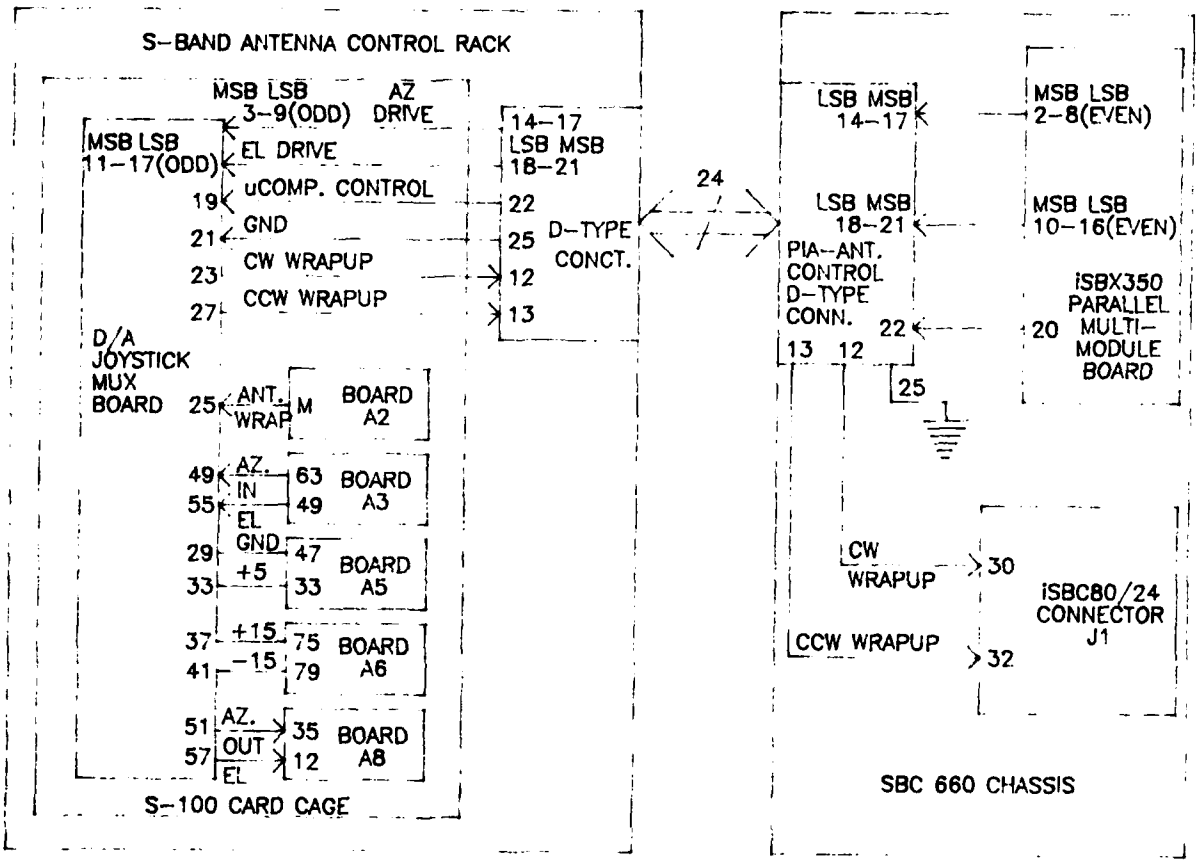


D/A - JOYSTICK MUX
BOARD SCHEMATIC
PAGE 1 OF 2



D/A – JOYSTICK MUX BOARD

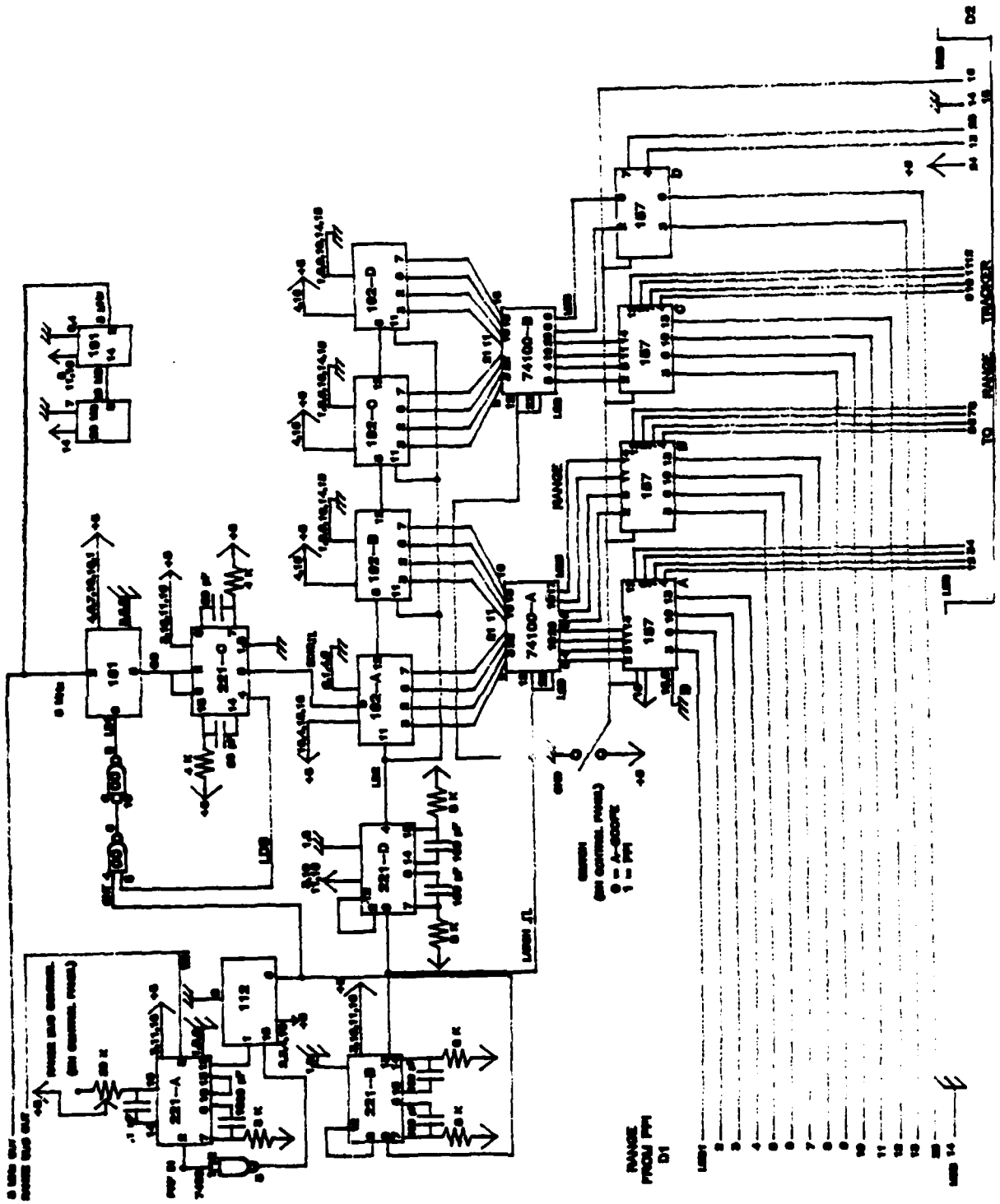
PAGE 2 OF 2



CABLING DIAGRAM
D/A – JOYSTICK MUX. BOARD

D/A–JOYSTICK MUX BOARD CABLING

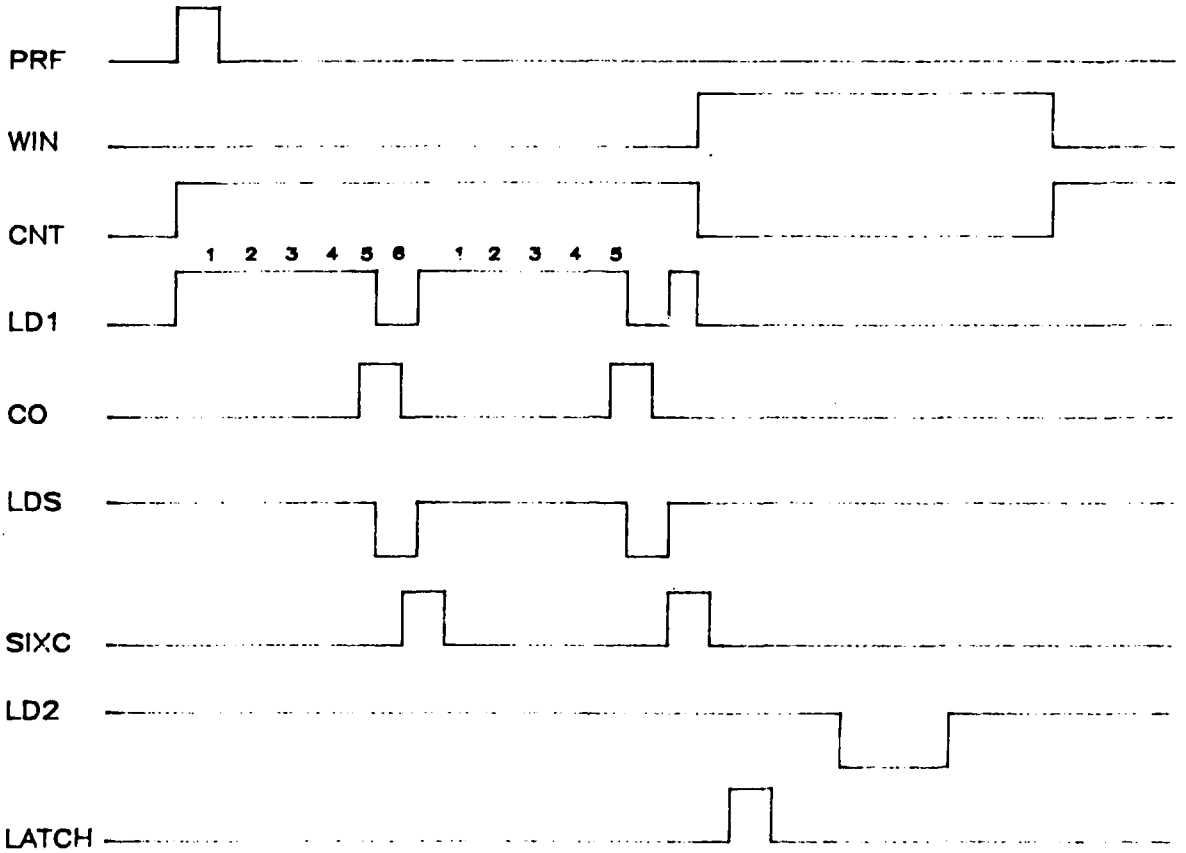
CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
EDGE	3–9 (ODD)	4 BIT AZ DRIVE INPUT (PIN3–MSB, PIN9–LSB)	PARALLEL MULTIMODULE PINS 2–8(EVEN) PORT B (F1) HO BITS
	11–17 (ODD)	4 BIT EL DRIVE INPUT (PIN11–MSB, PIN17–LSB)	PARALLEL MULTIMODULE PINS 10–16(EVEN) PART B (F1) LO BITS
	19	uPROC CONTROL BIT	PARALLEL MULTIMODULE PORT C (F2) BIT 2
	21	GND	uPROC GND
	23	CLOCKWISE WRAPUP SIGNAL	ISBC 80/24 (J2–30) PORT EA–BIT 6
	25	WRAPUP INFO FROM S–BAND DRIVE +V–CW, –V–CCW	A2–M
	27	CCW WRAPUP SIGNAL	ISBC 80/24 (J2–32) PORT EA–BIT 7
	29	GND	S–BAND CONTROLLER GND A5–47
	33	+5	A5–33
	37	+15	A6–75
	41	–15	A6–79
	45	–5	N/C
	49	JOYSTICK AZ DRIVE INPUT	A3–63
	51	AZ DRIVE OUTPUT	A8–35
	55	JOYSTICK EL DRIVE INPUT	A3–49
	57	EL DRIVE OUTPUT	A8–12



INITIAL RANGE CONTROL BOARD
SCHEMATIC

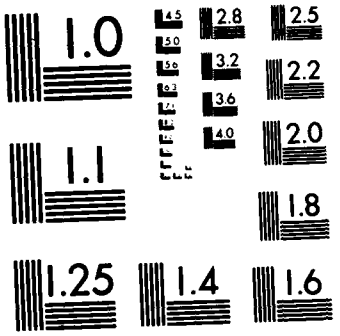
TIMING DIAGRAM

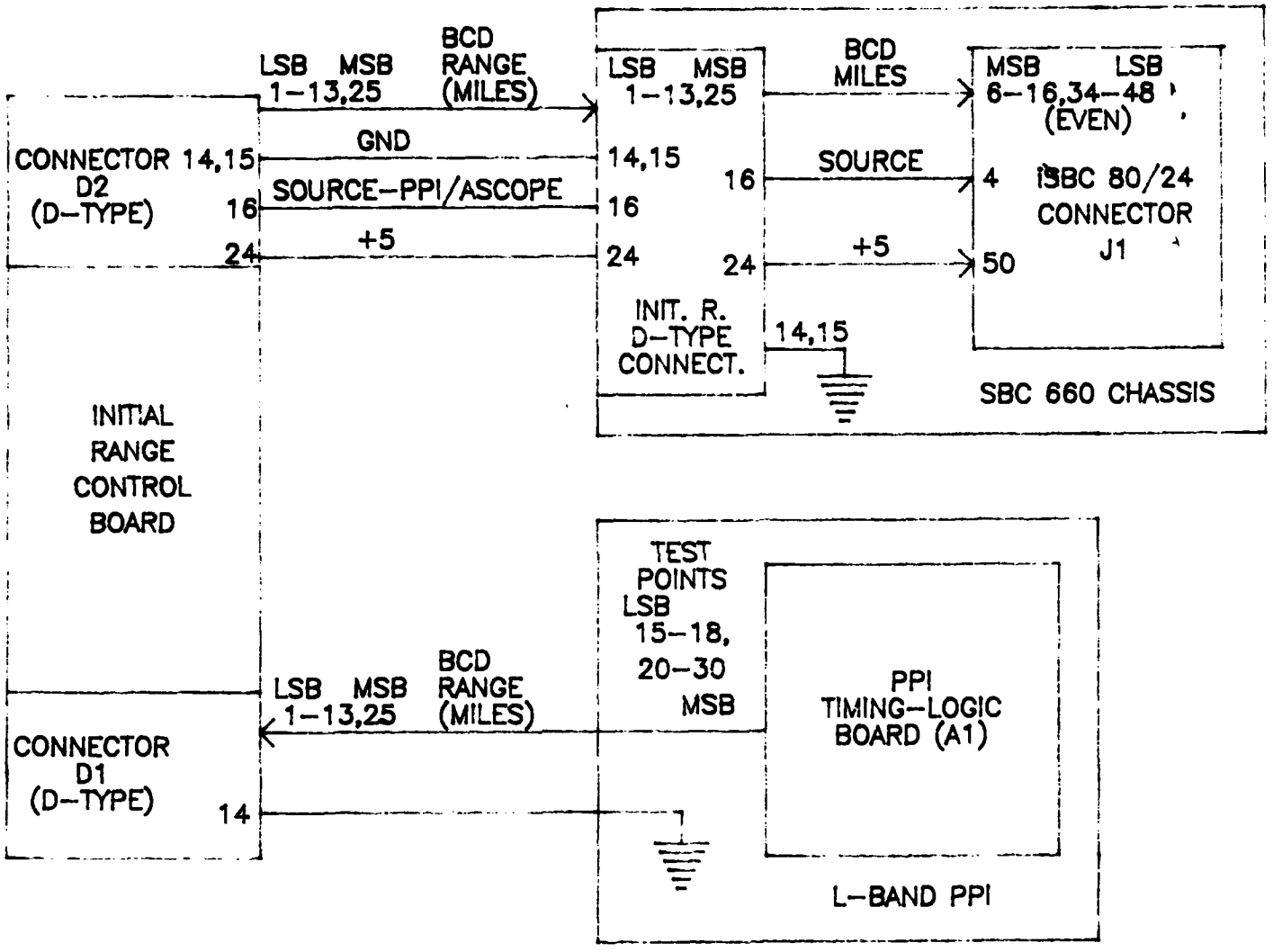
INITIAL–RANGE CONTROL BOARD



AD-A134 730 S-BAND RANGE TRACKER AND SURVEILLANCE LAB INTERFACE(U) 2/2
ROME AIR DEVELOPMENT CENTER GRIFFISS AFB NY B D BUSH
SEP 83 RADC-TR-83-213

UNCLASSIFIED F/G 17/9 NL

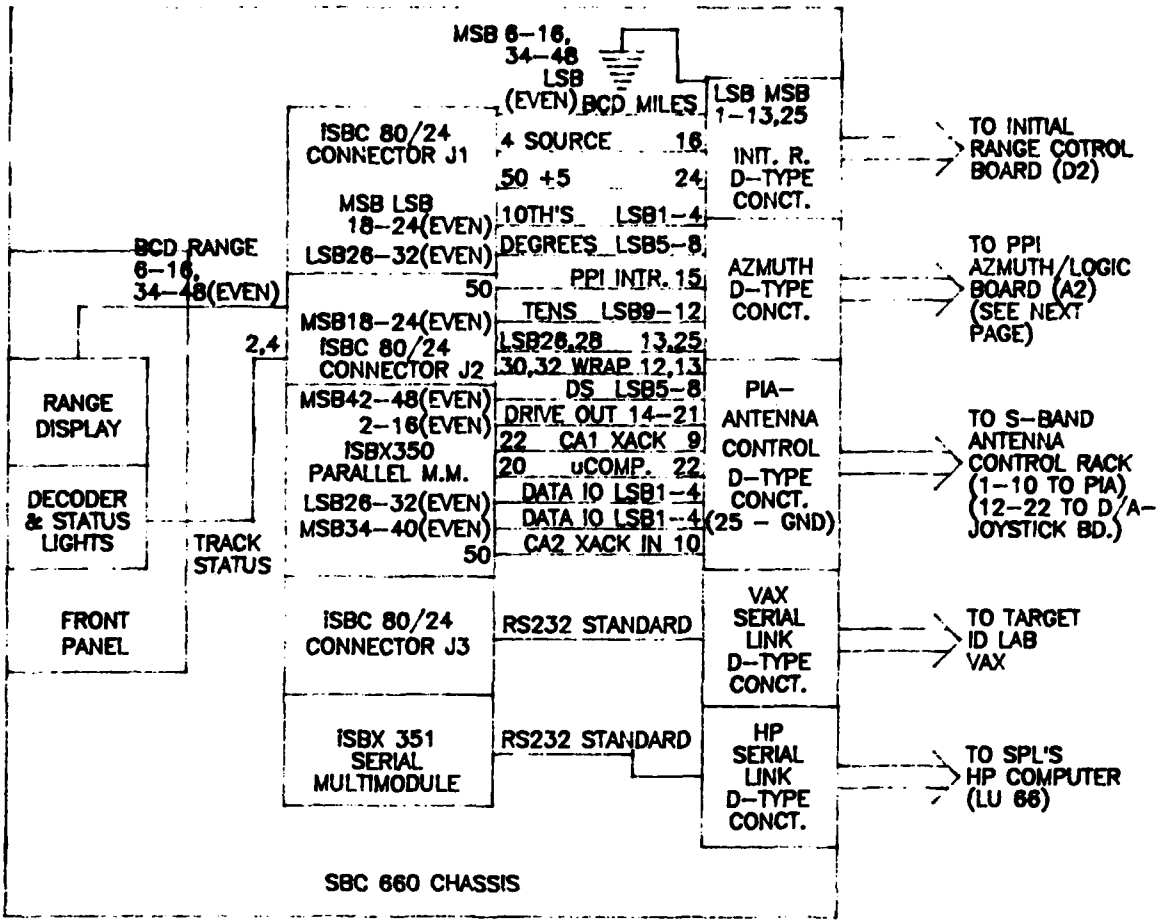




CABLING DIAGRAM
INITIAL RANGE CONTROL BOARD

CABLING IR CONTROL BOARD

CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
D1 – RANGE IN FROM PPI	1–4	BCD TENTHS OF MILES (1–LSB, 4–MSB)	PPI TIMING LOGIC BOARD (A1) TEST POINTS 15–18
	5–8	BCD MILES (5–LSB, 8–MSB)	PPI TIMING LOGIC BOARD (A1) TEST POINTS 20–23
	9–12	BCD TENS OF MILES (9–LSB, 12–MSB)	PPI TIMING LOGIC BOARD (A1) TEST POINTS 24–27
	13,25	BCD HUNDREDS OF MILES (25–MSB)	PPI TEST LOGIC BOARD (A1) TEST POINTS 28,30
	14	GND	PPI–GND
D2 – RANGE OUT	1–4	BCD TENTHS OF MILES (1–LSB, 4–MSB)	ISBC (80/24) JI(42–48(EVEN)) (48–LSB) PORT E4(LO)
	5–8	BCD MILES (5–LSB, 8–MSB)	ISBC(80/24) JI(34–40(EVEN)) (40–LSB) PORT E4(HO)
	9–12	BCD TENS OF MILES (9–LSB, 12–MSB)	ISBC(80/24) JI(10–16(EVEN)) (16–LSB) PORT E5(LO)
	13,25	BCD HUNDREDS OF MILES (25–MSB)	ISBC(80/24) JI(6,8) (6–MSB) PORT E5 BITS 4&5
	14,15	GND	uCOMPUTER GND
	24	+5	ISBC(80/24) JI(50)
	16	SOURCE PPI/A–SCOPE	ISBC(80/24) JI(4) PORT E5 BIT 6



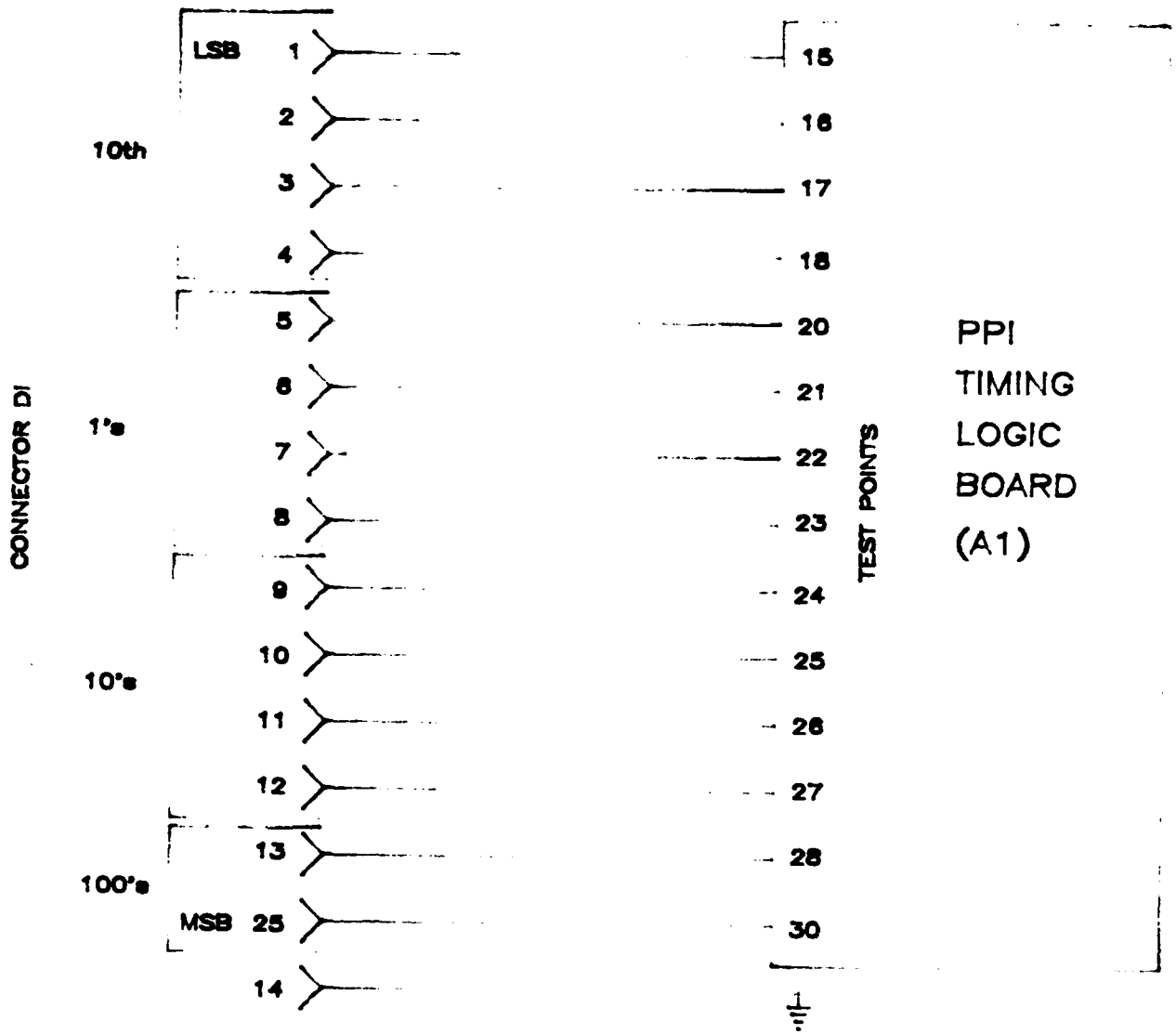
CABLING DIAGRAM
ISBC 80/24 & MULTIMODULE BOARDS

ISBC 80/24 MULTIMODULE CABLING

CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
P1	1–86	MULTIBUS STD.	RF BOARD & TC BOARD
P2	1–60	POWER BACKUP CONNECTOR	N/C
J1	1–49 (ODD)	GND	
	2	PORT E5 BIT 7 GROUND (UNUSED)	GND (J1–PIN 1)
	(EVEN)	(PORT E5 BIT0 – BIT6) INITIAL RANGE INFORMATION	SEE IR CONTROL BOARD CABLING
	18–24 (EVEN)	AZIMUTH INPUT FROM PPI BCD PORT E6 BIT0 – BIT3 TENTHS OF DEGREES (18–MSB, 24–LSB)	PPI–GND
	26–32 (EVEN)	PORT E6 BIT4 – BIT7 AZIMUTH INPUT FROM PPI– BCD DEGREES (26–LSB, 32–MSB)	PPI AZIMUTH LOGIC BOARD (A2) TEST POINTS 82–85 (82–MSB, 85–LSB)
	34–50 (EVEN)	PORT E4 INITIAL RANGE INFORMATION	SEE IR CONTROL BOARD CABLING
J2	1–49 (ODD)	GND	
	2,4	TRACKING STATUS CONTROL (PORT E9) 4,2 BIT 6&7 0,0 – STANDBY 0,1 – LOCKED 1,0 – COAST 1,1 – NO OUTPUT	TO DECODER & TRACKING STATUS LIGHTS ON FRONT PANEL
	6,8	BCD RANGE OUTPUT– HUNDREDS OF MILES (PORT E9 BIT 4&5) (8–LSB)	TO MSD OF BCD RANGE DISPLAY (FRONT PANEL)

ISBC 80/24 MULTIMODULE CABLING

CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
J2	10–16 (EVEN)	BCD RANGE OUTPUT TENS OF MILES (16–LSB, 10–MSB) (PORT E9 BIT0 – BIT3)	TO TENS DIGIT OF BCD RANGE DISPLAY (FRONT PANEL)
	18–24 (EVEN)	AZIMUTH INPUT FROM PPI BCD – TENS OF DEGREES (18–MSB, 24–LSB) (PORT EA BIT0 – BIT3)	PPI AZIMUTH LOGIC BOARD (A2) TEST POINTS 77,78,80,81 (77–MSB, 81–LSB)
	26,28	AZ INPUT FROM PPI BCD– HUNDREDS OF DEGREES (26–LSB) (PORT EA BIT4, BIT5)	PPI ALB (A2) TEST POINTS 76 & 75 (75–MSB)
	30–32	S–BAND WRAPUP IN (PORT EA BIT 6&7)	SEE D/A JOYSTICK MUX BOARD CABLING
	34–48 (EVEN)	BCD RANGE OUTPUT MILES & TENTHS OF MILES (34–MSB, 48–LSB) (PORT E8 BIT0 – BIT7)	TO BCD RANGE DISPLAY ON FRONT PANEL
	50	PPI INTERRUPT IN	TARGET ID BUTTON ON L–BAND PPI
J3		RS232 INTERFACE (PORT EC)	VAX COMPUTER– TARGET–ID LAB
SERIAL MULTIMODULE ISBX 351		RS232 INTERFACE (PORT C0)	HP COMPUTER SIGNAL PROCESSING LAB

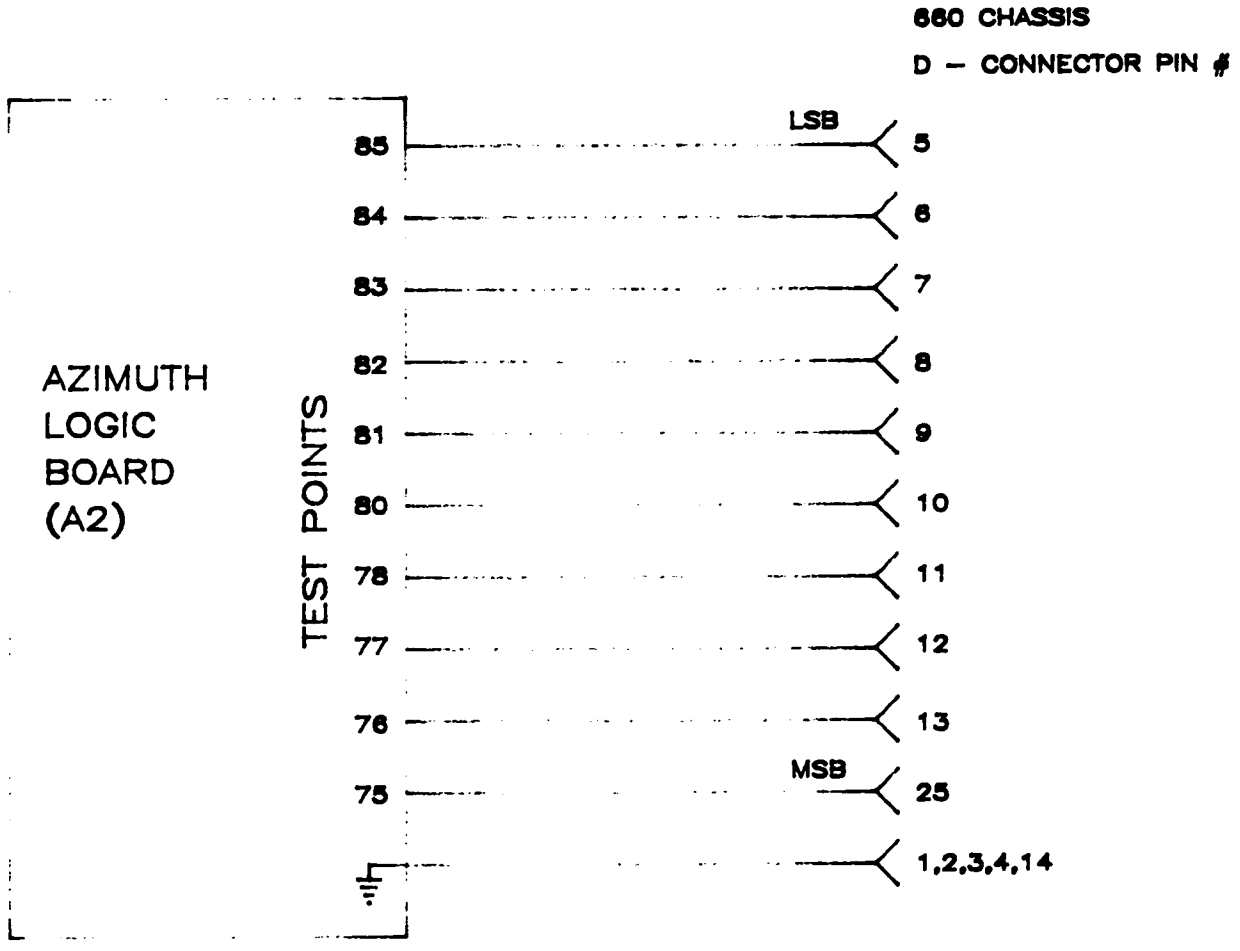

BCD RANGE INFO FROM PPI
CONNECTOR DI
LSB
1
2
10th
3
4
5
6
1's
7
8
9
10
10's
11
12
13
100's
MSB
25
14
TEST POINTS
15
16
17
18
20
21
22
23
24
25
26
27
28
30
PPI
TIMING
LOGIC
BOARD
(A1)

ISBC 80/24 MULTIMODULE CABLING

CONNECTOR	PIN(S)	DESCRIPTION	DESTINATION CONNECTOR & PIN #
PARALLEL MULTIMODULE ISBX 350	1–49 (ODD)	GND	
	2–16 (EVEN)	S–BAND ANTENNA DRIVE OUTPUT (PORT F1)	SEE D/A JOYSTICK MUX BOARD CABLING
	22	CA1 HANDSHAKE LINE TO PIA (PORT F2 BIT1)	TO CA1 INPUT TO PIA INTERFACE
	20	ANTENNA DRIVE– uCOMPUTER CONTROL LINE (PORT F2 BIT2)	SEE D/A JOYSTICK MUX BOARD CABLING
	26–32 (EVEN)	DATA INPUT FROM PIA (26–LSB, 32–MSB) (PORT F2 BIT 4–7)	PIA DATA I/O (DR) P.M.M. PINS(34–40(LSB))
	34–40 (EVEN)	4 BIT DATA OUTPUT TO PIA (34–MSB, 40–LSB) (PORT F0 BIT 4–7)	PIA DATA I/O P.M.M. PINS(26–32(MSB))
	42–48 (EVEN)	DATA SELECT OUTPUT TO PIA (PORT F0 BIT0–3) (42–MSB, 48–LSB)	DATA SELECT (DS) INPUT TO PIA
	50	CA2 HANDSHAKE IN FROM PIA (PIA INTERRUPT)	PIA CA2 OUTPUT

BCD AZIMUTH DATA FROM PPI

(AN/UPA – 62C)



APPENDIX B

SOFTWARE LISTINGS

For simplicity sake the software listings have been omitted from this report. The listing of the 8085 assembly program consists of approximately 2000 lines of source code and the HP software consists of about 1000 FORTRAN statements. A complete listing of all the software would have produced an appendix comparable in length to that of the report which describes it. I will therefore provide the file names on the Surveillance Lab's Hewlett Packard computer in which these source listings can be accessed should there be a need to do so.

FILE S80.09 - 8085 SOFTWARE ROUTINES: SLUAT RETRV PIAIO WRTRG

FILE S80.08 - ALL OTHER 8085 ROUTINES

FILE GATES - ALL HP FORTRAN ROUTINES

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.